



## INFORMATITZACIÓ DE L'ARXIU D'HISTÒRIES CLÍNIQUES D'UN HOSPITAL

Memòria del Projecte Fi de Carrera  
d'Enginyeria en Informàtica

realitzat per

**David González Buisán**

i dirigit per

**Pilar Gómez Sánchez**

Bellaterra, 17 de setembre de 2009

### **Agraïments:**

A la Pilar Gómez, la meva tutora de projecte, per guiar-me i aconsellar-me en tot moment, al llarg de la realització del projecte.  
A José Carlos Ruíz i Ángel Luis López, de HP, i a Enrique Miguel Martín, de HP-CDS, per permetre'm presentar aquest PFC.  
A tots els meus companys de feina, per donar-me un cop de mà quan ho he necessitat.  
I, sobretot, als meus pares, ja que sense la seva insistència ara no seria on sóc, i a la Carol, ja que sense ella no seria qui sóc.





El sotasignat, **Angel Luis López López**  
de l'empresa, **Hewlett-Packard**

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat en l'empresa sota la seva supervisió, mitjançant conveni signat amb la Universitat Autònoma de Barcelona

Així mateix, l'empresa en té coneixement i dóna el vist-i-plau al contingut que es detalla en aquesta memòria.

Signat: **Angel Luis López López**

Sabadell, 16 de setembre de 2009



El sotasignat, **Pilar Gómez Sánchez**

Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en

I per tal que consti firma la present.

Signat: **Pilar Gómez Sánchez**

Bellaterra, 17 de setembre de 2009

## **Índex**

<b>Capítol 1</b>	<b>Introducció.....</b>	<b>7</b>
1.1	Presentació .....	7
1.2	Motivacions per a realitzar aquest projecte.....	8
1.3	Motivacions personals.....	9
1.4	Objectius principals .....	9
1.5	Estat d'art .....	10
1.6	Estructura de la memòria .....	11
<b>Capítol 2</b>	<b>Estudi de Viabilitat.....</b>	<b>13</b>
2.1	Recursos .....	13
2.1.1	Recursos humans.....	13
2.1.2	Recursos hardware .....	13
2.1.3	Recursos software.....	14
2.2	Avaluació de costos .....	15
2.3	Organització del projecte .....	15
2.3.1	Estratègia de desenvolupament .....	15
2.3.2	Metodologia de desenvolupament.....	16
2.3.3	Característiques a tenir en compte.....	16
2.4	Planificació del projecte .....	16
2.5	Conclusions.....	20
<b>Capítol 3</b>	<b>Anàlisi de Requeriments .....</b>	<b>21</b>
3.1	Requeriments funcionals.....	21
3.1.1	Tasques AHC .....	21
3.1.2	Consulta de Llistats .....	25
3.1.3	Perfils d'Usuari .....	25
3.2	Requeriments no funcionals.....	25
3.2.1	Restriccions de Rendiment.....	25
3.2.2	Restriccions de Disseny .....	26
3.2.3	Restriccions d'Accés .....	26
3.2.4	Conclusions .....	26
3.3	Especificacions Tècniques.....	27
<b>Capítol 4</b>	<b>Necessitats que ha de cobrir el mòdul AHC.</b>	<b>28</b>
4.1	Introducció.....	28
4.2	Circuit de Consultes Externes.....	29
4.3	Circuit d'Admissions.....	31
4.4	Circuit d'Estudis .....	31
4.5	Circuit de la resta de motius de petició .....	32
4.6	Recerca de dossiers no prestats.....	32
4.7	Sol·licitud de dossiers a CLS .....	32
4.8	Enviament de dossiers a CLS.....	34
4.9	Circuit de Recuperació de dossiers .....	35

## **Capítol 5 Taules necessàries pel mòdul AHC ..... 36**

5.1 Introducció.....	36
5.2 Taules d'HCIS necessàries per al mòdul AHC .....	36
5.3 Taules pròpies del mòdul AHC .....	40
5.4 Conclusions.....	48

## **Capítol 6 Implementació del mòdul AHC..... 49**

6.1 Introducció.....	49
6.1.1 Java .....	49
6.1.2 JavaServer Pages (JSP) .....	50
6.2 Estil de programació .....	50
6.2.1 Variables i mètodes .....	51
6.2.2 Classes .....	51
6.3 Estructura del codi .....	51
6.3.1 Formats dels blocs de codi.....	51
6.3.2 Comentaris .....	52
6.4 Classes per interactuar amb la base de dades .....	53
6.4.1 El concepte de Datamanager .....	53
6.4.2 Classes per treballar amb les dades .....	54
6.5 JSPs i classes de les pantalles .....	67
6.5.1 El fitxer actions.properties .....	67
6.5.2 Les pantalles .....	67
6.5.3 Els panells .....	68
6.5.4 Les classes Ajax.....	69
6.6 Les accions .....	69
6.6.1 Les accions que actuen sobre dossiers .....	70
6.6.2 Les accions dels manteniments .....	79

## **Capítol 7 Proves realitzades ..... 81**

## **Capítol 8 Conclusions ..... 82**

## **Capítol 9 Bibliografia..... 83**

9.1 Referències i articles.....	83
9.2 Altres web .....	83
9.3 Llibres.....	83

## **Annex ..... 84**

A.1 Pantalla de Gestió de Dossiers .....	84
A.2 Pantalla de Gestió de Peticions.....	86
A.3 Pantalla de Gestió de Préstecs .....	88

## **Capítol 1 Introducció**

### **1.1 Presentació**

En l'actualitat, **Hewlett-Packard**, a partir d'ara **HP**, està fent a la **Corporació Sanitària Parc Taulí** (Hospital de Sabadell), d'aquí en endavant **CSPT**, la implantació de l'aplicació que està desenvolupant per a hospitals. Aquesta aplicació rep el nom d'**HCIS**, acrònim de "Health Care Information System".

El projecte que es presenta a continuació consisteix en el desenvolupament de la part de l'aplicació **HCIS** que s'encarrega d'informatitzar l'**Arxiu d'Històries Clíniques**, des d'aquest moment **AHC**.

Actualment, la **CSPT** té el seu propi sistema informàtic, però aquest consta de dues eines per a la gestió clínica de pacients, un Sistema d'Informació Clínic (**SIC**) per a la consulta de proves diagnòstiques i elaboració d'informes d'alta, una Estació de Treball Clínic (**ETC**) per a gestionar la ubicació de pacients, informes d'alta, petició i consulta de proves, així com un Sistema d'Informació Organitzatiu (**SIO**), per a la gestió organitzativa dels clients i de la facturació de l'entitat. Aquest sistema informàtic, però, és força antic, a més de ser molt complex degut a l'elevat nombre de diferents eines que el formen, fet pel que la Corporació va decidir canviar-lo per tal de modernitzar-se i, sobretot, aconseguir tenir una única aplicació que integri totes les anteriors.

**HCIS**, però, és una aplicació que encara no està finalitzada, motiu pel que s'han anat implantant diferents mòduls poc a poc, entenent per mòdul d'**HCIS** tot el referent a una àrea determinada, com poden ser Urgències, Admissions, Consultes Externes, etc. Òbviament, la idea de la **CSPT** és que els seus professionals acabin fent servir única i exclusivament **HCIS**, però, per la raó que s'acaba d'esmentar, això encara no és possible. En l'actualitat, conviuen tots dos sistemes, **HCIS** i el sistema propi de la **CSPT**, usant un sistema de missatgeria per a comunicar-se entre ells.

D'aquesta manera, el projecte que té **HP** a la **CSPT** consta de dues parts força diferenciades. La primera és la implantació dels mòduls ja existents a **HCIS**. I la segona és el desenvolupament i implantació dels mòduls restants per tal de poder aconseguir que **HCIS** esdevingui una aplicació completa per a gestionar un hospital i englobar totes i cadascuna de les diferents eines de les que consta el sistema informàtic de la **CSPT**.

D'altra banda, **HP** pretén que **HCIS** sigui una aplicació que es pugui instal·lar a molts més hospitals. De fet, hi ha diversos projectes similars, com poden ser el de l'Hospital Sant Joan de Déu, el de Mútua de Terrassa o el de l'Hospital de Sant Boi, entre d'altres. És per aquest motiu que s'ha de mirar que el desenvolupament dels nous mòduls sigui el més estàndard possible, ja que posteriorment s'haurà d'implantar en altres hospitals. De la mateixa manera, en els altres hospitals es desenvolupen nous mòduls que després s'implantaran a la **CSPT**.



Per aquesta raó, aquest projecte no consistirà en fer una aplicació a mida per a què treballi l'AHC de la CSPT.

Però és necessària la informatització de l'Arxiu d'Històries Clíniques de la CSPT dintre de l'aplicació HCIS, desenvolupada per HP, perquè els usuaris de l'AHC deixin d'utilitzar l'aplicació SIO per passar a fer servir HCIS.

El principal avantatge que aportarà aquest nou mòdul de HCIS serà aconseguir un entorn molt més fàcil d'utilitzar i d'entendre per a l'usuari, ja que l'aplicació desenvolupada per HP és una aplicació web, on l'ús del ratolí és molt important, fent de la fàcil interacció de l'usuari amb l'aplicació un dels punts forts. A més, es poden realitzar diferents accions des de la mateixa pàgina. Per la seva banda, el SIO és una aplicació de menús i formularis, havent-se de fer servir el teclat numèric per moure's entre els diferents menús i formularis. D'aquesta forma, els principals beneficiats d'aquest canvi seran els usuaris, ja que passaran a tenir una aplicació molt més intuïtiva i senzilla d'usar. A més, tot i que no sigui el més comú, de vegades han d'usar alguna altra eina del sistema informàtic antic, com, per exemple, el SIC. Així s'aconseguirà que el personal de l'arxiu clínic només hagi d'utilitzar una aplicació informàtica.

Per altra banda, també s'aconseguirà tenir, a llarg termini, a tots els usuaris de la CSPT treballant amb la mateixa aplicació informàtica.

## **1.2 Motivacions per a realitzar aquest projecte**

Com s'acaba de comentar, a la CSPT hi havia un sistema informàtic compost per diverses eines.

El que es vol acabar aconseguint és que HCIS sigui l'única aplicació informàtica que faci servir el personal de la CSPT. Per aquest motiu, cada cop més mòduls de l'aplicació es van desenvolupant en HCIS, deixant-se d'utilitzar les corresponents eines dintre del sistema informàtic de la CSPT.

En l'actualitat conviuen les dues aplicacions informàtiques, existint un sistema de missatgeria per a comunicar-se entre elles.

Per tal de continuar amb el projecte que engloba aquest, implantació total d'HCIS a la CSPT, s'ha decidit continuar amb el desenvolupament del mòdul d'AHC perquè és un mòdul força tancat, en el sentit que els usuaris són bastant específics i es dediquen exclusivament a realitzar la feina de l'arxiu clínic. D'aquesta manera s'evita, en la major mesura possible, tenir usuaris que hagin de treballar amb les dues aplicacions.

Així doncs, amb la realització d'aquest projecte, s'aconseguirà que un bon nombre de treballadors de la corporació passin a treballar única i exclusivament amb HCIS.

## 1.3 Motivacions personals

La idea de realitzar aquest desenvolupament com a projecte de final de carrera ha sorgit a partir de la xerrada que s'ha donat als estudiants al començament del curs acadèmic. En ella s'ha animat a l'estudiant a fer les seves pròpies propostes i, en cas que s'estigui treballant, a realitzar la proposta d'un projecte real per a la seva empresa.

Aquest darrer cas ha suposat un fort al·licient per a presentar la proposta, ja que d'aquesta manera es pot treballar sobre un projecte de veritat, que un cop acabat estarà en funcionament al món real.

Finalment, el fet de tractar-se d'un projecte similar, tot i que de major escala, que altres vistos al llarg de la carrera, també ha suposat un bon incentiu a l'hora de voler realitzar aquest projecte, ja que permet poder aplicar coneixements assolits al llarg dels anys d'estudi, cosa que és un dels objectius del projecte final de carrera.

## 1.4 Objectius principals

Tal i com ja s'ha comentat, el principal objectiu d'aquest projecte és desenvolupar el mòdul d'HCIS que s'encarregui de gestionar l'arxiu clínic de la CSPT, i que aquest sigui el més estàndard possible per a poder-se implantar en altres hospitals.

Al tractar-se d'un hospital que rep moltes visites al llarg del dia, la quantitat d'informació que es belluga és molt gran, pel que l'usuari ha d'estar interactuant constantment amb l'aplicació. És per això que és molt important que el mòdul AHC sigui el més senzill possible d'utilitzar.

Finalment, com el desenvolupament del mòdul d'AHC es pot considerar un projecte que està inclòs dins d'un altre projecte més gran (implantació d'HCIS a la CSPT), també persegueix assolir l'objectiu d'acabar aconseguint que el personal de la CSPT només treballi amb HCIS, deixant d'usar el sistema informàtic de la CSPT.

Per acabar, solsament comentar que, al tractar-se d'un projecte força llarg, s'ha hagut de dividir en tasques més petites per tal de facilitar-ne la seva realització i així assolir d'una forma més senzilla els objectius establerts. Aquestes tasques s'han repartit des del mes de febrer del 2.008 fins al mes de juny del 2.009, tal i com es podrà observar en l'Estudi de viabilitat, capítol 2 d'aquesta memòria, on es fa una explicació més detallada de cadascuna de les diferents tasques.

## 1.5 Estat d'art

L'estat d'art és un anàlisi de la situació actual del mercat entorn al projecte a desenvolupar.

A aquestes alçades, resulta evident que les noves tecnologies estan canviant els nostres hàbits de conducta i cada cop més la nostra quotidianitat està sent ocupada per diferents tecnologies, que ens fan la vida més còmoda.

El sector sanitari és un dels més importants i sensibles dintre de l'economia espanyola, ja que és un element clau del benestar social. Per aquest motiu, el sector de la sanitat s'enfronta a nombrosos reptes, i les Tecnologies de la Informació i les Comunicacions (TIC) obren un ampli ventall de possibilitats de renovació i millora. És per aquest motiu que el Govern Espanyol va posar en marxa el programa "*Sanidad en Línea*", havent-li destinat 140 milions d'euros per part del Ministeri d'Indústria, que amb les aportacions del Ministeri de Sanitat i de les comunitats autònomes, va ascendir a 252 milions d'euros. [1]

Per la seva part, la Unió Europea incentiva i desenvolupa cinc àrees per garantir la cobertura i continuïtat sanitària en tot l'àmbit europeu. Aquestes àrees són: la *Història Clínica Electrònica* (HCE), la *Recepta Electrònica*, la *Mobilitat*, la *Telemedicina*, i la *Picture Archive Communication System* (PACS) o emmagatzemament digital dels exàmens de diagnòstic del pacient. [2]

Amb tot això, l'aplicació de les TIC en l'àmbit de la sanitat ofereix nombroses possibilitats, ja que, pràcticament en tots els àmbits on la sanitat pot actuar, les TIC poden jugar un paper molt important per a obtenir una millora de la productivitat. Alguns dels aspectes en els que les TIC poden influir notablement són els següents:

- **Millora de relacions:** l'ús adequat de les TIC pot permetre atendre determinats pacients als professionals sense necessitat de desplaçar-se, o també podrà millorar la relació entre professionals al poder realitzar consultes on-line.
- **Optimització de processos:** l'impacte de les TIC en els processos propis de la sanitat (com poden ser els processos assistencials o terapèutics) és enorme, ja que l'activitat sanitària comporta la col·laboració d'un gran nombre de professionals i d'altres entitats.
- **Recolzament a professionals:** és facilita la investigació i l'accés a la informació i el seu intercanvi. A més, es disposa de més informació per millorar la presa de decisions.
- **Gestió de la informació:** tota activitat sanitària genera molta informació. L'emmagatzemament, tractament i distribució de la mateixa ocupa una gran quantitat d'espai i de recursos. La utilització de les TIC no només permet reduir recursos, sinó que també ajuda a la reducció d'errors i a l'accés a la informació des de qualsevol lloc.

Per tots aquests motius, cada vegada són més els hospitals que es troben immersos en projectes per implantar software sanitari, per tal de modernitzar-se i aconseguir oferir els millors serveis possibles.

Degut a que és un sector emergent, cada cop hi ha més empreses que estan apostant pel sector de la sanitat i que s'estan encarregant de realitzar projectes d'aquest tipus (desenvolupament i implantació de software per a hospitals). Una d'elles és HP, que compta amb el seu producte HCIS, que és un entorn amb múltiples funcionalitats que cobreixen els fluxos de treball de les organitzacions sanitàries. A part de la CSPT, altres hospitals com la *Mútua de Terrassa*[3] o l'*Hospital de Sant Joan de Déu*[4] també es troben immersos en projectes per implantar el producte HCIS.

D'altres empreses, com poden ser *T-Systems* o *Siemens*, també estan desenvolupant i implantant el seu propi software. Per exemple, T-Systems, juntament amb Net2u\_, ha desenvolupat una eina de gestió per a la Xarxa Sanitària Aragonesa[5]. Per la seva banda, Siemens ha implantat el seu software *Selene* en diversos hospitals, com pot ser l'hospital *San Agustín*, de la localitat asturiana d'Avilès[6].

Finalment, comentar que aquest és el software que s'està implantant en grans hospitals, però hi ha d'altres empreses més petites, com poden ser *Savac* o *COSTAISA*, que també tenen el seu propi producte, que s'està implantant en hospitals de menor mida. Per exemple, a l'*Hospital d'Igualada* o al *Sant Jaume* d'Olot[7] treballen amb el producte de Savac, mentre que la *Mútua Castellonense*[8] fa servir com a sistema de gestió el producte *Chaman*, de l'empresa COSTAISA.

## 1.6 Estructura de la memòria

Per a acabar la introducció, en aquest darrer punt s'explica com està estructurada la memòria.

En primer lloc, en aquest capítol, es realitza una breu introducció, on s'inclouen la idea general del projecte, els principals objectius que s'han d'assolir, així com les motivacions que han portat a realitzar-lo. A més també hi ha l'estat d'art, on s'analitza què hi ha a l'actualitat al mercat.

Al segon capítol, es realitza un estudi de viabilitat del projecte, analitzant quins són els seus recursos necessaris i fent una planificació en el temps. Això permetrà decidir si el projecte és o no viable des del punt de vista tècnic i econòmic.

En el tercer capítol es fa un anàlisi dels requeriments de l'aplicació, partint de la base del problema, tot especificant els requeriments necessaris.

El quart capítol ja tracta quins són els circuits de treball que se segueixen a un Arxiu d'Històries Clíniques, per tal de saber quines són les necessitats que ha de cobrir el mòdul AHC.

Al següent capítol, el cinquè, ja es comencen a detallar parts que formen el mòdul AHC, més concretament el model de dades, explicant quines són les taules que es fan servir i quina és la seva finalitat.

Finalment, s'explica la implementació del mòdul AHC al capítol sis, per acabar amb les proves i tests que s'han fet al capítol set i extreure'n les conclusions al capítol vuit.

## **Capítol 2 Estudi de Viabilitat**

En aquest capítol es tracten tots els requeriments que ha de complir qualsevol projecte per tal de que es pugui considerar viable i es pugui dur a terme. S'ha de tenir en compte que un projecte ha de ser viable a nivell tècnic i econòmic, que és el que es tractarà de demostrar tot seguit.

### **2.1 Recursos**

Per a portar a terme qualsevol tipus de projecte són necessaris diferents tipus de recursos, sense els quals seria impossible la seva realització.

A continuació es detallaran els recursos necessaris per a la realització del present projecte.

#### **2.1.1 Recursos humans**

Per a la realització del projecte general (implantació de l'aplicació HCIS), HP consta amb un equip professional format per un *Cap de Projecte*, un *Coordinador de Projecte*, dos *Analistes*, dos *Programadors* i un *Integrator*.

Per al desenvolupament del mòdul AHC, però, només és necessari un programador, amb la corresponent supervisió per part del Coordinador i Cap de Projecte.

A l'apartat de planificació es detalla el temps dedicat a la realització del projecte.

#### **2.1.2 Recursos hardware**

Aquest apartat fa referència als recursos de maquinari necessaris per al desenvolupament del projecte.

Bàsicament, el hardware necessari per a la realització del projecte és el servidor on estarà l'aplicació. Aquest és un servidor Jboss en la seva versió 4.2.3.

Jboss és un servidor d'aplicacions J2EE que està implementat en Java, fet que fa possible que pugui ser utilitzat en qualsevol sistema operatiu.

Tot el referent al manteniment i configuració del servidor, però, correrà a càrrec del personal de l'àrea de Sistemes de la CSPT.

L'aplicació estarà desplegada en dos servidors diferents, sent el que s'acaba de descriure d'ús exclusiu per al mòdul d'AHC. L'altre servidor és on treballen la resta d'usuaris d'HCIS.

El motiu de tenir dos servidors és tenir el menor nombre possible d'usuaris a cada servidor, per tal de no carregar-lo en excés, fent que el rendiment sigui el millor possible.

En l'actualitat, l'únic servidor que està en funcionament és un servidor Orion, que és un altre servidor d'aplicacions J2EE diferent. El motiu pel que el nou servidor serà Jboss és perquè en d'altres projectes s'ha obtingut un millor rendiment amb servidors d'aquest tipus que no pas amb servidors Orion.

La intenció és acabar migrant l'actual servidor a Jboss en un futur, però això no es tractarà en aquest projecte.

### 2.1.3 Recursos software

En aquest apartat es farà referència al software necessari per a la realització d'aquest projecte.

En primer lloc, el sistema operatiu que hi ha en aquests sistemes és *Linux*. L'objectiu d'aquest projecte no és canviar el sistema operatiu, ja que Linux dona molts bons resultats, per tant, no es farà cap modificació al respecte.

Els llenguatges de programació fets servir per a la programació dels diferents mòduls d'HCIS són *Java (J2EE)*, *Javascript* i *HTML*. Així, s'ha decidit que per al mòdul AHC es facin servir els mateixos llenguatges que per a la resta de l'aplicació. La versió de JDK de Java feta servir és la *JDK 1.6.0\_07*. [9]

L'eina de desenvolupament és *IntelliJ IDEA 7.0.4* [10], ja que les versions anteriors no suporten el JDK 1.6.0\_07

Pel que fa al sistema gestor de base de dades, el que fa servir HCIS és *Oracle 10g* [11], sent *Toad for Oracle 9.5* [12] l'aplicació informàtica usada per a interactuar amb la base de dades.

Altres productes software fets servir són *SmartCVS 6* [13] i *IReport 2.0.5*. [14]

CVS (Concurrent Versions System) és una aplicació informàtica que implementa, mitjançant una arquitectura client-servidor, un sistema de control de versions. El CVS emmagatzema els arxius font de cada versió, així com un historial de canvis i un registre dels usuaris que han accedit a cada fitxer. *SmartCVS* és un client de CVS.

Finalment, *JasperReports* és un producte software, també desenvolupat en Java, per a la creació de llistats i d'informes que dona la possibilitat de mostrar els resultats per pantalla, enviar-los a la impressora o generar documents PDF, HTML, XLS, CSV o XML. *IReport* és una eina gràfica per a editar informes creats amb JasperReports.

## **2.2 Avaluació de costos**

En aquest apartat es tracta de fer una valoració dels costos que pot suposar el dur a terme aquest projecte, tenint en compte els recursos que s'acaben d'esmentar.

Pel que fa al cost, tant del hardware com de les llicències necessàries per als diferents productes software, aquest corre a càrrec de la CSPT. Al ser la CSPT l'empresa encarregada de fer la despesa econòmica, no s'ha pogut tenir accés a les quantitats emprades per a adquirir tant la maquinària com el software mencionat en el punt anterior.

A més d'això, també s'ha de tenir en compte el cost dels recursos humans, pel que fa al sou que reben els treballadors, així com la quantitat que ha de pagar la CSPT a HP per comptar amb aquests recursos humans. Aquestes dades, però, tampoc es facilitaran.

## **2.3 Organització del projecte**

Perquè un projecte es pugui dur a terme correctament, aquest s'ha d'organitzar prèviament per tal de tenir ben clar, fins i tot abans de començar a realitzar-lo, quins són els passos que s'han de seguir.

### **2.3.1 Estratègia de desenvolupament**

L'organització que s'ha decidit per a aquest projecte és la següent:

1. Anàlisi de la situació a tractar.
2. Cerca d'informació sobre projectes similars realitzats anteriorment.
3. Obtenir informació exacta sobre què es vol que faci l'aplicació.
4. Establiment dels requeriments funcionals del sistema.
5. Establiment dels requeriments no funcionals del sistema.
6. Estudi de possibles alternatives.
7. Planificació de les diferents etapes del projecte.
8. Adquisició de tots els recursos software i hardware necessaris per a la implementació i funcionament del projecte.
9. Realització del projecte.
10. Seguiment i supervisió del funcionament de l'aplicació un cop feta la implantació en producció.



### 2.3.2 Metodologia de desenvolupament

Dins del món de l'Enginyeria del Software trobem que hi ha diferents models de desenvolupament. Els més estudiats al llarg de la carrera són el Cicle Lineal, el Model Evolutiu i el Prototipus.

El primer model queda descartat degut al fet que implica que fins que una fase no s'ha acabat no es pot començar amb l'altra. Això implica que el projecte ha d'estar compost per una sèrie de fases consecutives i aquest no és el nostre cas.

El model de Prototipus no és el més apropiat, ja que no hi ha una aplicació força semblant en la que es pugui basar el desenvolupament d'aquest projecte.

Per tant, tan sols queda el Model Evolutiu. Aquest és el més apropiat, ja que el mòdul AHC es va elaborant gradualment i, tot i que també es desglossa el projecte en etapes, n'hi ha algunes que es poden realitzar al marge de les altres sense afectar als temps establerts per a la realització del projecte.

### 2.3.3 Característiques a tenir en compte

1. Al finalitzar el desenvolupament del mòdul, s'hauran de fer uns cursos de formació per als usuaris de l'arxiu clínic de la CSPT, per tal que n'aprenguin el seu correcte funcionament.
2. Caldrà que es facin moltes proves per tal d'assegurar que el mòdul AHC d'HCIS funciona correctament.
3. Seria força interessant fer un recull de tots els problemes que vagin sorgint al llarg del projecte, així com de quins passos s'han seguit per a solucionar-los, per tal de tenir una guia per a possibles projectes similars.

## 2.4 Planificació del projecte

En les següents figures on es mostra la planificació de les tasques (Figura 2.1.) i el diagrama de Gantt (Figura 2.2.), es pot observar la planificació temporal del projecte, així com les diferents tasques que el formen i les precedències que hi ha entre elles.

Com es pot observar, hi ha 7 grans tasques que conformen el projecte, cadascuna amb les seves corresponents subtasques, per tal de facilitar la feina.

La primera gran tasca és la que consisteix en la *Gestió de dossiers*, entenent per dossier la carpeta (o carpetes) física on es guarden els informes de la història clínica del pacient.

La següent és la que s'encarrega de fer la *Gestió de peticions*, ja que per a què un professional pugui rebre el dossier del pacient, primer haurà d'existir una sol·licitud per tal que hi hagi constància.

La tercera tasca important és la *Gestió de préstecs*. Aquesta tasca aconseguirà controlar a quins professionals se'ls ha deixat cada dossier.

Per altra banda, per problemes de volum, la CSPT consta de tres arxius, motiu pel qual també s'hauran de gestionar els *Traslats de dossiers* entre els diferents arxius.

Una altra tasca important serà l'*Elaboració de llistats*, ja que a l'arxiu clínic també es treballa molt amb llistats impresos en paper.

Un cop s'hagin realitzat cadascuna d'aquestes tasques, s'haurà de fer una de les tasques més importants en qualsevol projecte, la de *Proves i tests*.

Finalment, s'hauran de fer els *Cursos de formació per als usuaris*, per tal que el personal de l'arxiu clínic de la CSPT estigui capacitat per a treballar amb el mòdul AHC d'HCIS.

	Task Name	Duration	Start	Finish	Predecessors
1	<b>Informatzació de l'Arxiu d'Històries Clíniques</b>	<b>77,75 days</b>	<b>Mon 02/02/09</b>	<b>Tue 26/05/09</b>	
2	<b>Gestió de dossiers</b>	<b>37,75 days</b>	<b>Mon 02/02/09</b>	<b>Wed 25/03/09</b>	
3	Crear dossier	2 days	Mon 02/02/09	Tue 03/02/09	
4	Modificar dossier	3 days	Wed 04/02/09	Fri 06/02/09	3
5	Eliminar dossier	2 days	Mon 09/02/09	Tue 10/02/09	3
6	Crear volum	2 days	Wed 11/02/09	Thu 12/02/09	5
7	Modificar volum	2 days	Fri 13/02/09	Mon 16/02/09	6
8	Eliminar volum	1 day	Tue 17/02/09	Tue 17/02/09	6
9	Prestar dossier	1 day	Wed 25/03/09	Wed 25/03/09	22
10	Recuperar dossier	1 day	Wed 25/03/09	Wed 25/03/09	22
11	Imprimir etiquetes	3 days	Wed 18/02/09	Fri 20/02/09	8
12	Consultar visites per dossier	1 day	Mon 23/02/09	Mon 23/02/09	3
13	Consultar intervencions quirúrgiques per dos	1 day	Mon 23/02/09	Mon 23/02/09	3
14	<b>Gestió de peticions de dossiers</b>	<b>20,75 days</b>	<b>Tue 24/02/09</b>	<b>Tue 24/03/09</b>	
15	Crear petició	2 days	Tue 24/02/09	Wed 25/02/09	8
16	Modificar petició	3 days	Thu 26/02/09	Mon 02/03/09	15
17	Eliminar petició	1 day	Tue 03/03/09	Tue 03/03/09	15
18	Preparar peticions	3 days	Wed 04/03/09	Mon 09/03/09	15
19	Prestar peticions	4 days	Tue 10/03/09	Fri 13/03/09	18
20	Recuperar peticions	3 days	Mon 16/03/09	Wed 18/03/09	19
21	Peticions amb l'arxiu extern (CLS)	2 days	Thu 19/03/09	Fri 20/03/09	20
22	Recepció peticions arxiu extern (CLS)	2 days	Mon 23/03/09	Tue 24/03/09	21
23	<b>Gestió de préstecs de dossiers</b>	<b>9 days</b>	<b>Thu 26/03/09</b>	<b>Tue 07/04/09</b>	
24	Crear préstec	3 days	Thu 26/03/09	Mon 30/03/09	22
25	Rectificar préstec	2 days	Tue 31/03/09	Wed 01/04/09	24
26	Modificar préstec	2 days	Thu 02/04/09	Fri 03/04/09	25
27	Eliminar préstec	2 days	Mon 06/04/09	Tue 07/04/09	24
28	<b>Traspàs de dossiers</b>	<b>6 days</b>	<b>Wed 08/04/09</b>	<b>Fri 17/04/09</b>	
29	Enviar dossiers a CLS	2 days	Wed 08/04/09	Thu 09/04/09	22
30	Rebre resposta del CLS	2 days	Tue 14/04/09	Wed 15/04/09	29
31	Traspasar dossiers entre ART i AR2	2 days	Thu 16/04/09	Fri 17/04/09	3
32	<b>Creació de llistats</b>	<b>14 days</b>	<b>Mon 20/04/09</b>	<b>Fri 08/05/09</b>	
33	Llistats d'informació de dossiers	4 days	Mon 20/04/09	Thu 23/04/09	8
34	Llistats d'informació de peticions	5 days	Fri 24/04/09	Thu 30/04/09	14
35	Llistats d'informació de préstecs	5 days	Mon 04/05/09	Fri 08/05/09	23
36	Proves i Tests	9 days	Tue 12/05/09	Fri 22/05/09	32
37	<b>Formació usuaris</b>	<b>2 days</b>	<b>Mon 25/05/09</b>	<b>Tue 26/05/09</b>	
38	Formació usuaris "administradors"	1 day	Mon 25/05/09	Mon 25/05/09	36
39	Formació usuaris "normals"	1 day	Tue 26/05/09	Tue 26/05/09	38

Figura 2.1. Tasques del projecte

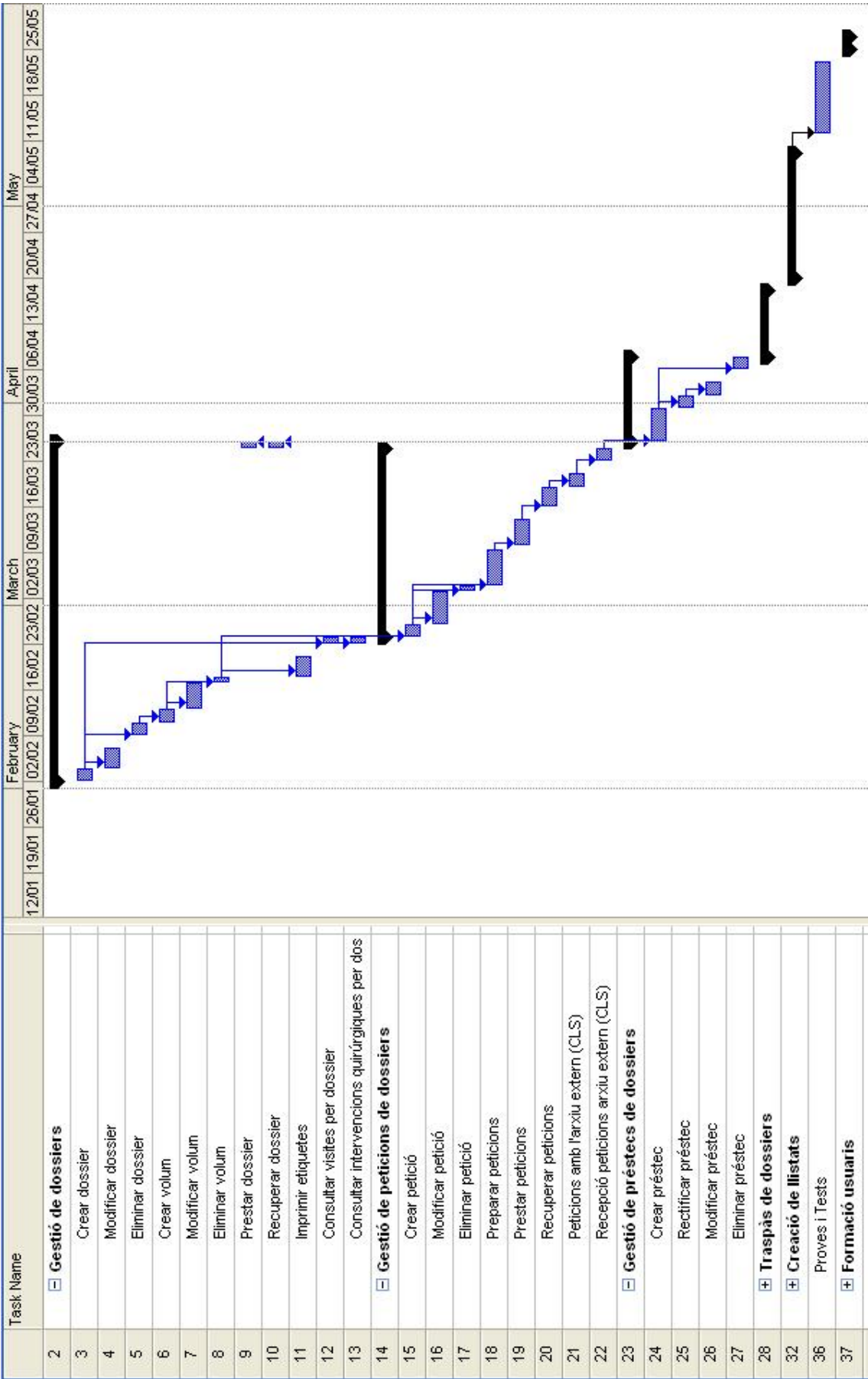


Figura 2.2. Planificació temporal del projecte complet

## 2.5 Conclusions

Després d'analitzar els costos i els avantatges i inconvenients que tindria aquest projecte, es considera viable la seva realització ja que s'obtindran diversos beneficis. Per una banda, amb aquesta aplicació s'aconsegueixen importants beneficis pels usuaris de l'arxiu clínic, ja que s'aconseguirà que puguin treballar amb una sola eina i que aquesta sigui molt més intuïtiva i senzilla de fer servir que l'anterior sistema informàtic.

A més, també s'ha vist que el projecte és totalment viable en els dos aspectes bàsics, ja que tècnicament es disposen de totes les eines necessàries i el cost econòmic ja es va negociar en el seu moment.

Finalment, la realització d'aquest projecte permet ampliar i millorar coneixements sobre la gestió d'un hospital i, més en concret, d'un arxiu clínic.

Per tant, tenint en compte els avantatges i els inconvenients, el projecte es pot considerar totalment viable.

## Capítol 3 Anàlisi de Requeriments

El conjunt d'idees que ajuden a desenvolupar el software és el que es coneix com a requeriments. Els que fan referència a la funcionalitat que l'usuari donarà al software resultant són els que es diuen funcionals, mentre que els que s'anomenen no funcionals són aquells que són restriccions imposades pel client o pel mateix problema i que afecten al disseny.

### 3.1 Requeriments funcionals

Els requeriments funcionals s'extreuen de les diferents tasques que porten a terme a l'AHC. Aquestes tasques es detallen a continuació:

#### 3.1.1 Tasques AHC

##### 3.1.1.1 Gestió de Dossiers

A la següent figura (Figura 3.1.) es mostra el diagrama de casos d'ús de totes les accions que pot realitzar un usuari des de la pantalla de *Gestió de Dossiers*.

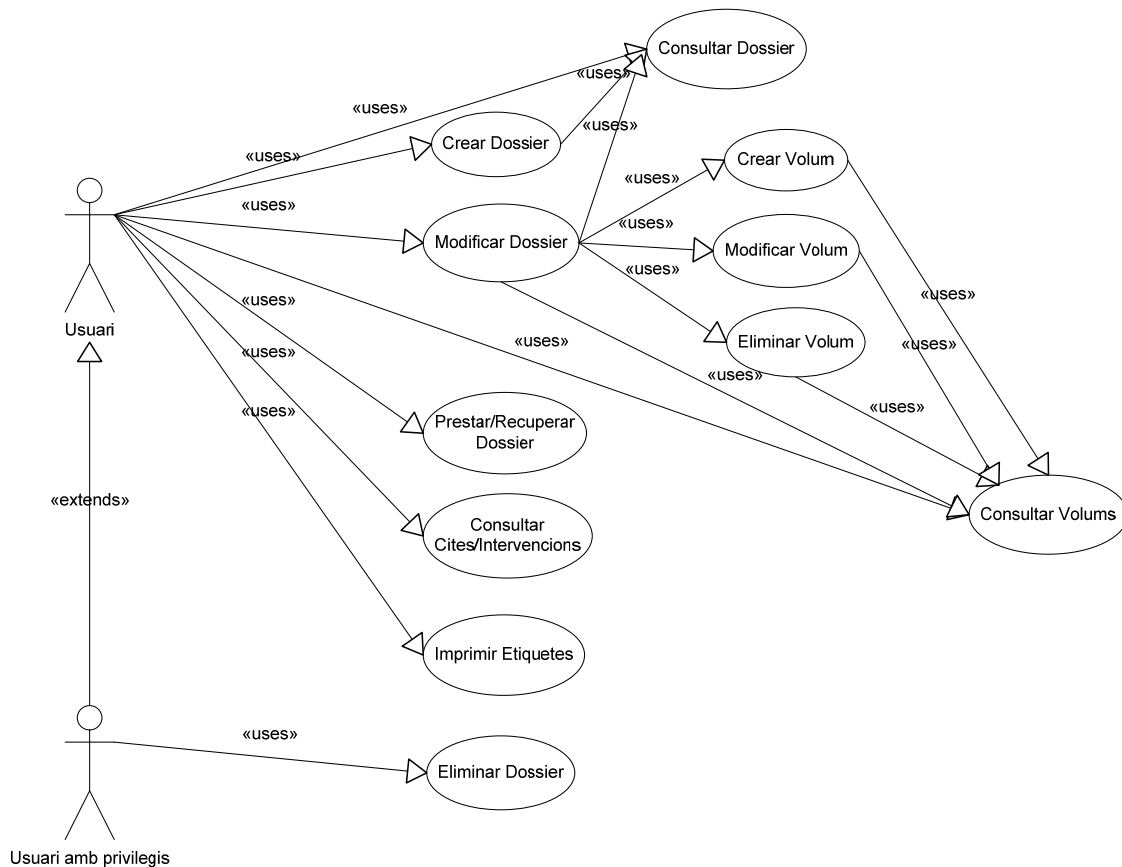


Figura 3.1. Diagrama de casos d'ús de la Gestió de Dossiers

Com es pot veure al diagrama, des de la gestió de dossiers, l'usuari ha de poder:

- Crear un nou dossier
- Consultar les dades d'un dossier
- Modificar les dades d'un dossier
- Eliminar un dossier <sup>(\*)</sup>
- Crear un nou volum a un dossier existent
- Consultar les dades dels volums d'un dossier
- Modificar les dades dels volums d'un dossier
- Eliminar un volum de un dossier <sup>(\*)</sup>
- Prestar un dossier
- Recuperar un dossier
- Imprimir etiquetes de codi de barres d'un dossier
- Consultar les visites de consultes externes del pacient al que pertany el dossier
- Consultar les intervencions quirúrgiques del pacient al que pertany el dossier

### 3.1.1.2 Gestió de Peticions

En el següent diagrama de casos d'ús (Figura 3.2.) es mostren totes les accions que es poden realitzar des de la *Gestió de Peticions*.

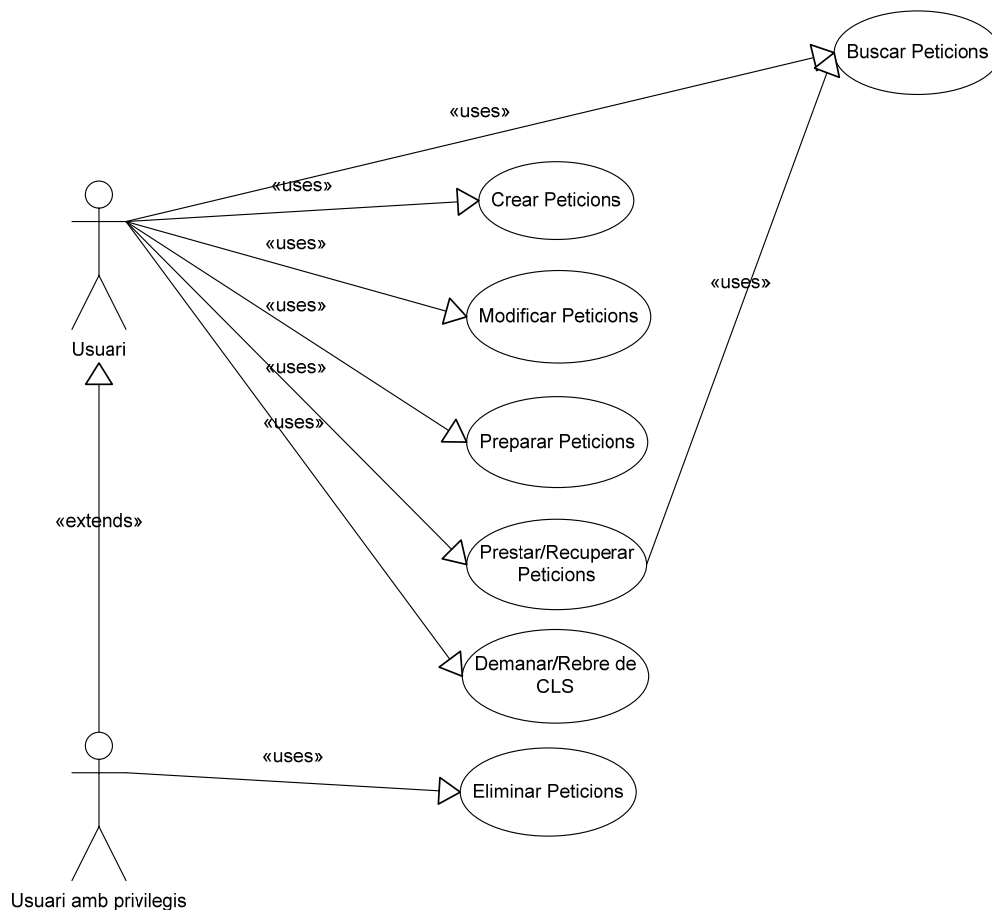


Figura 3.2. Diagrama de casos d'ús de la Gestió de Peticions

<sup>(\*)</sup> Per realitzar aquestes accions es necessiten permisos especials



Així, des de la gestió de peticions, l'usuari ha de poder:

- Crear peticions de dossier
- Consultar peticions de dossier
- Modificar peticions de dossier
- Eliminar peticions de dossier<sup>(\*)</sup>
- Preparar peticions de dossier
- Prestar peticions de dossier
- Recuperar peticions de dossier
- Realitzar peticions de dossier a l'arxiu extern (CLS)
- Rebre la resposta a les peticions realitzades a l'arxiu extern (CLS)

### 3.1.1.3 Gestió de Préstecs

Pel que fa a la *Gestió de Préstecs*, l'usuari ha de poder realitzar totes les accions que es mostren al diagrama representat a la Figura 3.3.

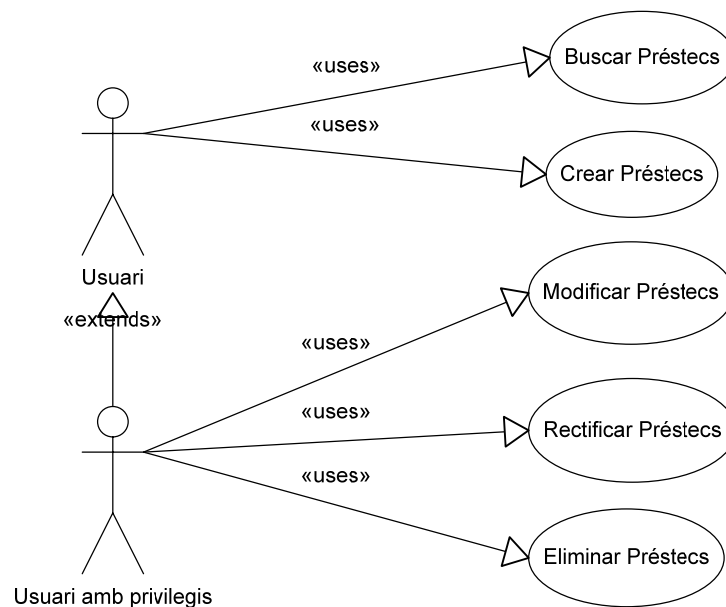


Figura 3.3. Diagrama de casos d'ús de la Gestió de Préstecs

És a dir, des de la gestió de préstecs, l'usuari ha de poder:

- Crear préstecs de dossier
- Consultar préstecs de dossier
- Modificar préstecs de dossier<sup>(\*)</sup>
- Rectificar préstecs de dossier<sup>(\*)</sup>
- Eliminar préstecs de dossier<sup>(\*)</sup>

### 3.1.1.4 Traspàs de Dossiers

Finalment, tal i com es veu a la Figura 3.4., des de la finestra de *Traspàs de Dossiers*, l'usuari ha de poder:

- Traspassar dossiers entre els arxius interns de la CSPT (ART i AR2)
- Enviar dossiers a l'arxiu extern (CLS)<sup>(\*)</sup>
- Rebre resposta sobre l'enviament de dossiers a l'arxiu extern (CLS)<sup>(\*)</sup>

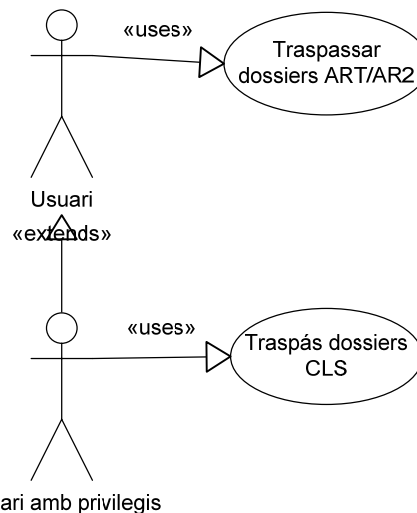


Figura 3.4. Diagrama de casos d'ús del Traspàs de Dossiers

### 3.1.2 Consulta de Llistats

A més de totes les funcionalitats especificades, l'usuari tindrà a la seva disposició una sèrie de llistats de treball per tal de facilitar-li la feina. Aquests llistats són importants, ja que hi ha part de la feina que s'ha de fer fora de l'arxiu, com per exemple, anar a recollir dossiers no retornats, i, en aquests casos, sense els llistats en paper, no es podria realitzar aquesta tasca.

### 3.1.3 Perfils d'Usuari

S'hauran de crear dos perfils diferents d'usuari, ja que n'hi haurà un que tindrà privilegis i permisos per a realitzar més tasques que l'altre. Per exemple, no tots els usuaris poden eliminar dossiers. La creació de perfils i l'assignació de privilegis i permisos als mateixos és quelcom que ja contempla HCIS en l'actualitat.

## 3.2 Requeriments no funcionals

Els principals requeriments que presenta el mòdul AHC són els següents.

### 3.2.1 Restriccions de Rendiment

És força important el fet d'obtenir temps de resposta el més baixos possible, per tal de facilitar la feina a l'usuari, ja que hi ha feines que són molt dinàmiques i necessiten molta interactivitat amb l'aplicació, requerint que les respostes siguin immediates. Un exemple pot ser el préstec de dossiers amb la pistola lectora de codi de barres, on es tenen una pila de dossiers i, per prestar-los, es va llegint cada etiqueta amb

la pistola. Si el temps de resposta és lent, s'hauran d'esperar entre préstecs, fent que el temps dedicat a aquesta tasca sigui molt més elevat del que hauria de ser.

Un dels principals punts crítics a l'hora d'oferir un bon rendiment són les consultes a la base de dades, pel que aquestes hauran de ser el més òptimes possible, per aconseguir que el seu temps de resposta sigui com més ràpid millor.

De la mateixa manera, també s'haurà d'intentar que el codi font de l'aplicació sigui el més ràpid possible.

Això, però, no només dependrà de la programació portada a terme, sinó que també hi tindran a veure altres factors, com pot ser el servidor. És per aquest motiu que s'ha decidit que el personal de l'Arxiu d'Històries Clíniques tindrà un servidor específic per a utilitzar l'aplicació.

### **3.2.2 Restriccions de Disseny**

Al ser un nou mòdul dintre d'una aplicació ja existent, òbviament, el disseny estarà restringit a que sigui el més similar possible a la resta de l'aplicació, per tal de donar-li el mateix aspecte a tots els diferents mòduls que formen HCIS.

### **3.2.3 Restriccions d'Accés**

HCIS és una aplicació que fan servir molts usuaris, però no tots han de tenir accés a totes les funcionalitats d'HCIS (de fet, només l'administrador en té), sinó que cada usuari solsament pot accedir a aquelles funcionalitats que necessita per a poder dur a terme la seva feina diària. Així, un usuari que treballi a Urgències, per exemple, no ha de tenir accés al mòdul AHC, ni a l'inrevés.

De la mateixa manera, no tots els usuaris d'arxiu tindran accés a totes les funcionalitats del mòdul AHC. Hi ha tasques més "delicades" que només podran executar determinades persones, com, per exemple, eliminar dossiers.

### **3.2.4 Conclusions**

Aquestes són les restriccions que incideixen en determinats aspectes de la qualitat del software.

Al tractar-se d'una aplicació que l'hauran de fer servir una gran diversitat d'usuaris, amb diferents nivells de coneixements informàtics, el mòdul AHC haurà de ser el més senzill possible, per tal de facilitar-ne la feina al personal.

S'ha de garantir que les diferents pantalles que disposi el mòdul AHC siguin el més intuïtives possible per als usuaris, i que només presentin la informació indispensable, per tal de no sobrecarregar en excés les pantalles.

### **3.3 Especificacions Tècniques**

Igual que ha passat amb les restriccions de disseny, al estar aquest mòdul inclòs dintre d'una altra aplicació, no s'ha hagut de prendre cap decisió a l'hora d'escollir llenguatges de programació, Sistema Operatiu, etc.

D'aquesta manera, tal i com es comenta a l'Estudi de Viabilitat, els llenguatges de programació fets servir per al desenvolupament del mòdul AHC d'HCIS són Java (J2EE), Javascript i HTML. Pel que fa al servidor, aquest serà un JBoss 4.2.3 amb un Sistema Operatiu Linux i Oracle 10g com a Gestor de Base de Dades.

Finalment, pel que fa als ordinadors dels usuaris, només cal que estiguin connectats a la xarxa interna de la Corporació per a poder accedir a l'aplicació, i que tinguin l'Internet Explorer, ja que aquest és l'únic explorador pel que HCIS garanteix el seu correcte funcionament.

## **Capítol 4 Necessitats que ha de cobrir el mòdul AHC**

Un cop s'ha fet un estudi de la viabilitat i els requeriments del projecte, en aquest capítol s'explica quin és el circuit de treball que segueix el personal de l'Arxiu Clínic, per tal de saber quines són les necessitats que ha de cobrir el mòdul AHC d'HCIS.

### **4.1 Introducció**

El primer que cal explicar és que, perquè es pugui prestar una història clínica, la persona que la necessita ha de fer una petició (o sol·licitud) de la mateixa, per tal que el personal de l'Arxiu s'assabenti de que hi ha algun professional que requereix una determinada història.

Les peticions de dossiers poden ser de dos tipus, o bé PROGRAMADES o bé ESPONTÀNIES.

S'entén per peticions PROGRAMADES aquelles que es generen automàticament, sense que l'usuari hagi de fer una sol·licitud explícita del dossier. Per exemple, quan es donen cites per algunes Consultes Externes, com pot ser una visita preoperatòria, ja s'ha de generar una petició de dossier sense fer res més que donar la cita.

Per altra banda, les peticions ESPONTÀNIES són aquelles que requereixen una sol·licitud explícita per part de la persona interessada en rebre el dossier. Un exemple pot ser un metge que està realitzant un Estudi sobre una determinada malaltia i vol històries clíniques de pacients que la pateixen per tal d'obtenir informació. En aquest cas, aquest metge haurà de fer manualment les peticions de dossier.

A més a més, també s'ha de saber que la principal classificació de les peticions de dossier es fa pel motiu de sol·licitud, és a dir, si la història s'ha sol·licitat perquè el pacient té una visita de Consultes Externes, o perquè ha ingressat, o perquè hi ha un metge que vol realitzar un Estudi amb la seva història, etc.

El motiu de sol·licitud que més peticions de dossier realitza és Consultes Externes, acaparant gairebé el 80% de les mateixes. És per aquesta raó que divideixen el personal en dos grups, el que s'encarrega de les Consultes Externes i el que es dedica a realitzar tasques que no tenen a veure amb aquestes.

D'altra banda, també s'ha de tenir en compte que, degut a l'elevat nombre d'històries clíniques que manega la CSPT, es disposa de tres arxius, dos interns i un extern. Dels dos interns, n'hi ha un que conté la gran majoria dels dossiers, que anomenen ART (Arxiu Taulí), i un de secundari anomenat AR2 (Arxiu Taulí 2), on només hi ha les històries, la numeració de les quals acaba en una parella de dígits determinats (terminació), per exemple, les que acaben en '00'. Per acabar, l'arxiu extern s'anomena CLS (Centre Logístic de Suport) i es troba a Santa Perpètua de la Mogoda.

En aquest arxiu hi ha les històries dels pacients que han mort i les històries que fa força temps que no es presten. Així, també s'ha de tenir en compte que s'envien històries d'un arxiu a un altre.

Finalment, també s'ha de considerar que la CSPT està composta per diferents edificis, motiu pel qual és molt important saber a quin edifici s'ha de portar cada dossier que s'ha de prestar.

## 4.2 Circuit de Consultes Externes

Com ja s'ha comentat, el gran gruix de la feina el comporten les Consultes Externes, motiu pel qual té un circuit especial.

En primer lloc, s'ha de tenir en compte que el treball de Consultes Externes es fa pel següent dia laborable, ja que, degut al gran volum de dossiers que s'ha de prestar, resultaria impossible fer aquesta feina pel mateix dia.

També és important saber que els dossiers amb sol·licitud de Consultes Externes es presten a AGENDES. Una agenda acostuma a pertànyer a un metge determinat, com pot ser l'agenda PIFOL, que correspon a les visites de Consultes Externes que realitza el doctor Joan Pi i Folgueres, però també pot correspondre a la causa de la visita, com pot ser l'agenda PREOP, que correspon a les visites preoperatòries.

Dit això, la primera acció que realitza el personal d'Arxiu és preparar les peticions de Consultes Externes del dia següent. L'acció de preparar consisteix en saber on es troba cada història que té una sol·licitud just en aquest moment. D'aquesta manera, se sap quines històries ja es troben a l'arxiu i, per tant, ja es poden començar a apilar per a realitzar el préstec de les mateixes, i quines es troben fora de l'arxiu i, conseqüentment, s'ha d'anar a buscar per a poder-les prestar. Els dossiers que estan fora de l'arxiu, incloses les que estan al CLS, s'anomenen EXTERIORS.

Un cop feta la preparació, s'imprimeixen dos llistats. El primer és el *Llistat de treball diari*, on apareixen totes les peticions de dossier, fins i tot les de dossiers exteriors, ordenades per edifici, agenda i terminació. El segon és el *Llistat d'exteriors*, on, com el seu nom indica, només apareixen les peticions de dossiers exteriors.

Degut al gruix del Llistat de treball diari, es reparteixen la feina entre moltes persones, encarregant-se cadascuna d'unes quantes agendes. El procediment que se segueix per cada agenda és, en primer lloc, anar a buscar tots els dossiers que s'han de prestar per a l'agenda concreta i agrupar-los i, una vegada fet això, llegir l'etiqueta de cada dossier amb una pistola lectora de codis de barres, per tal de que els dossiers quedin marcats com a prestats.

Quan ja s'han prestat, informàticament parlant, tots els dossiers, es porten físicament a la sala o consultori que pertorqui a cada un.

Pel que fa al Llistat d'exteriors, hi ha una persona encarregada d'anar a buscar cada dossier allà on es troba, per posteriorment prestar-lo informàticament, buscant manualment la petició que li correspon, i lliurar-lo allà on sigui necessària la història.

Per mirar de fer més entenedor aquest circuit, s'il·lustra a la següent figura (Figura 4.1.)

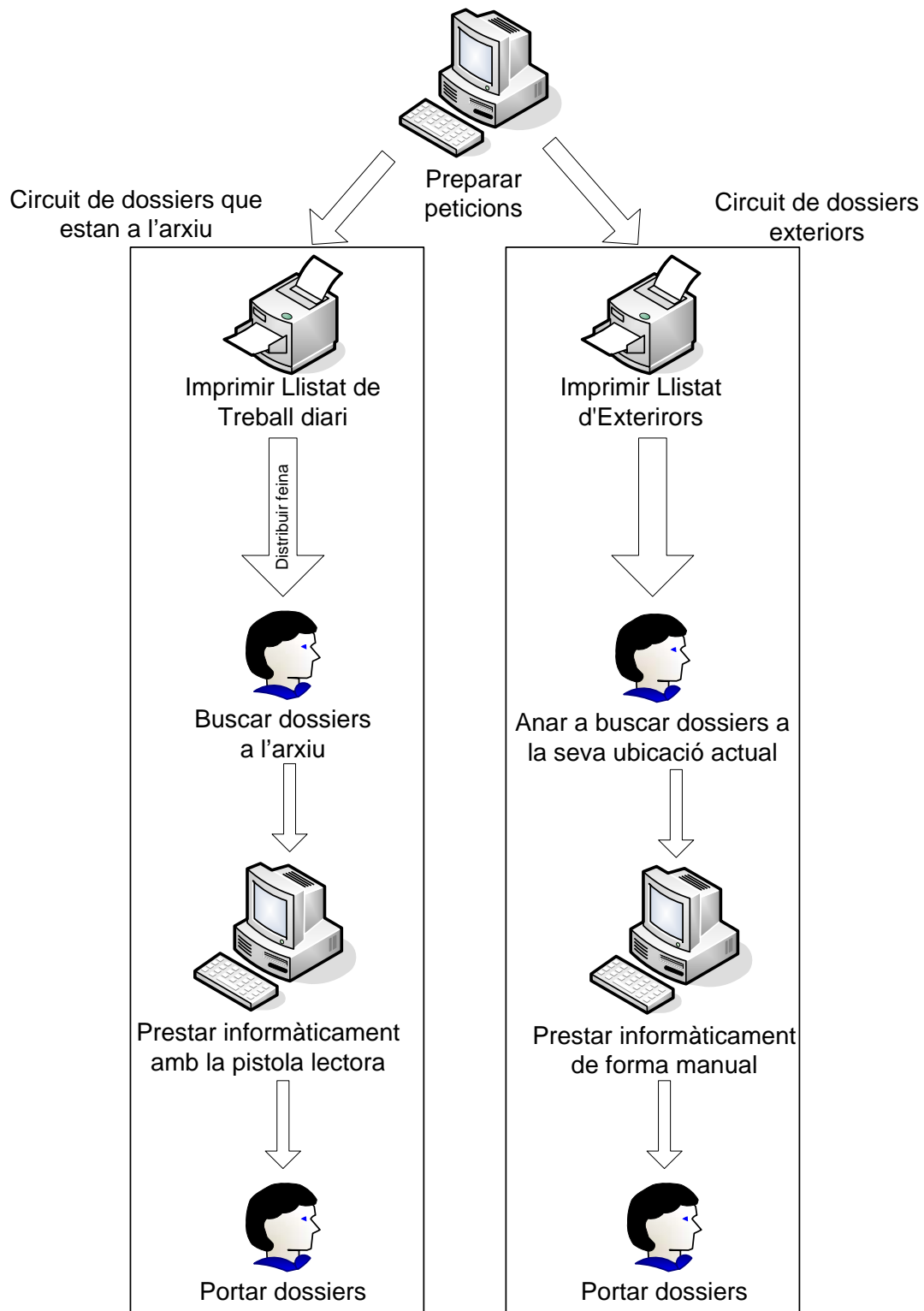


Figura 4.1. Circuit de Consultes Externes

## 4.3 Circuit d'Admissions

Per tal de prestar les històries dels pacients que estan hospitalitzats o han d'ingressar, es fan servir dos llistats.

El primer d'ells és el *Llistat de pacients ingressats que tenen la història a l'Arxiu*. En aquest llistat apareixen tots els pacients que ja es troben hospitalitzats i que, pel motiu que sigui, no se'ls ha portat el dossier a l'habitació. Una persona s'encarrega de buscar cadascun dels dossiers, prestar-los i portar-los al lloc que correspongui.

Per altra banda, hi ha el *Llistat de previsions d'ingrés*, on surten tots els pacients que està previst que ingressin al llarg de la tarda del mateix dia i del matí del següent. El personal d'arxiu s'encarrega de recopilar els dossiers corresponents i portar-los a l'habitació indicada, així, quan el pacient ingressi, el metge ja disposarà de la història clínica del malalt.

A la Figura 4.2. es veu gràficament el circuit, que és el mateix pels dos casos, canviant només el llistat que s'imprimeix i que, en el cas del primer llistat, es treu a primera hora del matí, mentre que el de previsions d'ingrés s'imprimeix a la tarda.

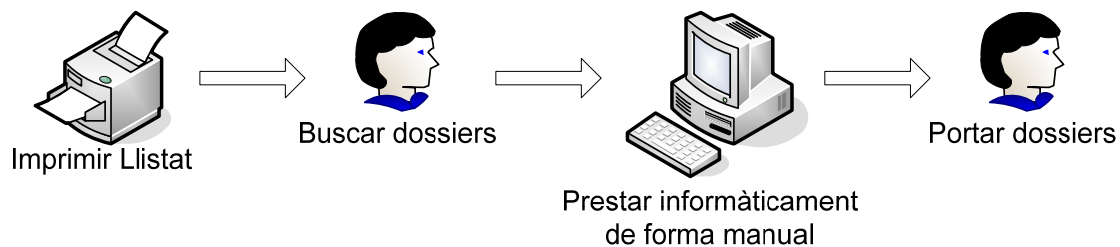


Figura 4.2. Circuit d'admissions

## 4.4 Circuit d'Estudis

El circuit que se segueix a l'hora de prestar dossiers per a realitzar Estudis és molt similar al de Consultes Externes, tot i que el volum de peticions és molt menor, motiu pel qual aquest circuit no es realitza a diari.

Primer es preparen les peticions d'històries sol·licitades per a realitzar Estudis, per posteriorment treure un llistat. Es busquen els dossiers demanats, ja estiguin a l'arxiu o siguin exteriors, i, un cop recopilats, es presten fent servir una pistola lectora de codis de barres. Un cop fet això es porten els dossiers a la persona que els ha demanat.



## 4.5 Circuit de la resta de motius de petició

Per la resta de motius de sol·licitud de dossiers, com poden ser Urgències, per a motiu Particular, per a una Auditoria, etc, les peticions es reben telefònicament o per correu electrònic. Es busca la història o històries demanades, es presten manualment i es porten allà on facin falta. La Figura 4.3. mostra visualment aquest circuit.

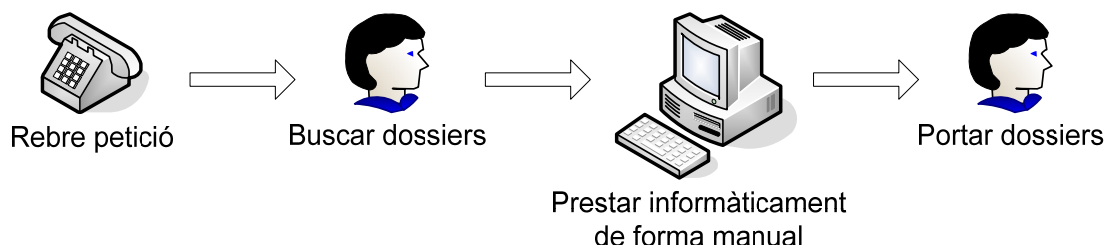


Figura 4.3. Circuit de la resta de motius de petició

## 4.6 Recerca de dossiers no prestats

Un cop ja s'ha finalitzat tot el procés de préstec de dossiers, es treu un altre llistat, anomenat *Llista Negra*, on apareixen els dossiers que haurien d'estar prestats i, pel motiu que sigui, encara no ho estan, independentment de quina sigui la seva ubicació actual.

El principal motiu acostuma a ser que la petició d'aquest dossier s'ha realitzat a última hora, quan ja s'havien imprès els llistats corresponents i ja s'havia començat amb els circuits habituals. Però també pot ser que perquè no es troba el dossier o simplement perquè al professional corresponent se l'hi ha passat per alt aquella història.

El procediment a seguir en aquest cas és el mateix que per les peticions que es reben via telefònica o per correu, és a dir, anar a buscar el dossier allà on sigui, prestar-lo manualment i, finalment, dur-lo allà on s'hagi requerit.

## 4.7 Sol·licitud de dossiers a CLS

Com ja s'ha comentat amb anterioritat, la CSPT consta de tres Arxius d'Històries Clíniques. Quan s'han de buscar dossiers que estan en qualsevol dels dos arxius interns (ART i AR2), no hi ha cap problema, ja que ambdós es troben en edificis de la Corporació. Però quan es necessita un dossier que es troba al CLS, s'ha de canviar la forma de treballar, ja que, al estar aquest arxiu en una altra localitat, no es pot disposar de la història clínica de manera immediata. Per aquest motiu, hi ha un circuit especial quan les històries requerides estan al CLS.

Aquest circuit consisteix en saber quines històries de les que s'han sol·licitat pels propers dies es troben a l'arxiu extern per mirar que, en el moment que s'hagin de prestar, ja estiguin a l'arxiu principal (ART).

Així, es treu un llistat on surten totes aquestes històries i s'envia un fitxer al personal del CLS indicant quines històries es necessiten. Quan el personal del CLS ja ha recopilat totes les històries, envia un altre fitxer de confirmació, indicant quins dossiers dels demanats s'han trobat. L'endemà següent es porten les històries des del CLS a l'arxiu ART, on es llegeixen les etiquetes amb una pistola lectora de codis de barres per que quedi constància de que aquests dossiers ja es troben a l'arxiu ART.

Aquest circuit es mostra a la Figura 4.4.

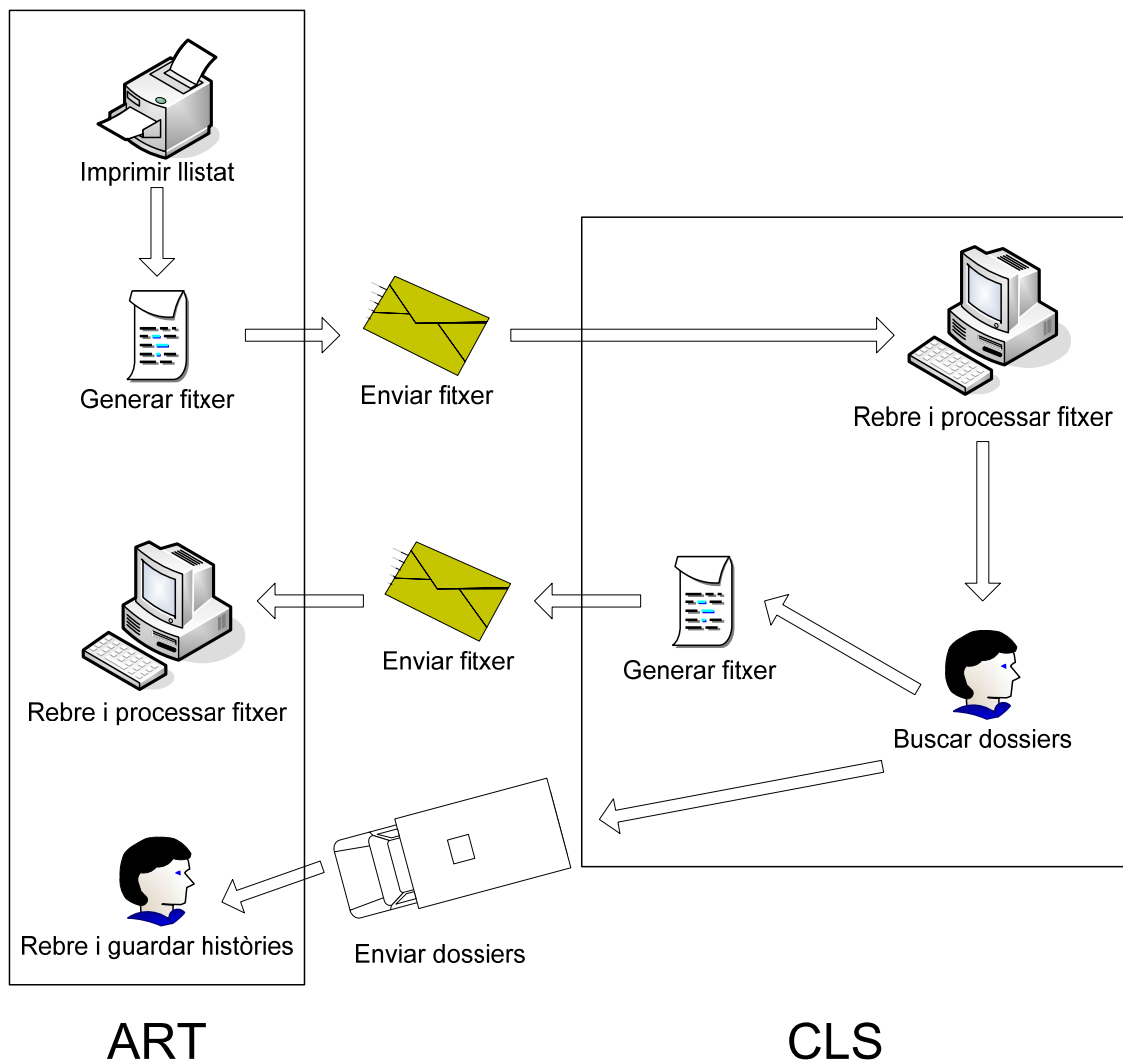


Figura 4.4. Sol·licitud de dossiers a CLS

## 4.8 Enviament de dossiers a CLS

Si s'han de sol·licitar dossiers a CLS, vol dir que prèviament s'han hagut d'enviar allà. Per fer-ho se segueix un circuit similar al de la sol·licitud.

El motiu que s'hagin d'enviar dossiers al CLS és simplement per manca d'espai. Els dossiers que s'envien a l'arxiu extern són, o bé els de pacients morts, o bé els dossiers que fa molt temps que no se sol·liciten (uns dos anys més o menys). Així, es consulta quins dossiers hi ha als arxius interns que són de pacients morts, i quins fa temps que no es presten, i se seleccionen per a enviar-los al CLS.

Un cop es tenen tots els dossiers, amb una pistola lectora de codis de barres es llegeixen les corresponents etiquetes per a que consti que s'enviaran al CLS. A la vegada, els dossiers dels pacients morts es marquen com a tals (s'anomenen ÈXITUS).

Quan ja s'ha acabat de llegir amb la pistola, es genera un altre fitxer per enviar al CLS, on consten tots els dossiers que s'enviaran.

Una vegada reben els dossiers al CLS, responen enviant un nou fitxer que conté les històries que han rebut. D'aquesta manera ja es pot considerar que aquests dossiers estan al CLS.

La Figura 4.5. mostra gràficament el circuit d'enviament de dossiers al CLS.

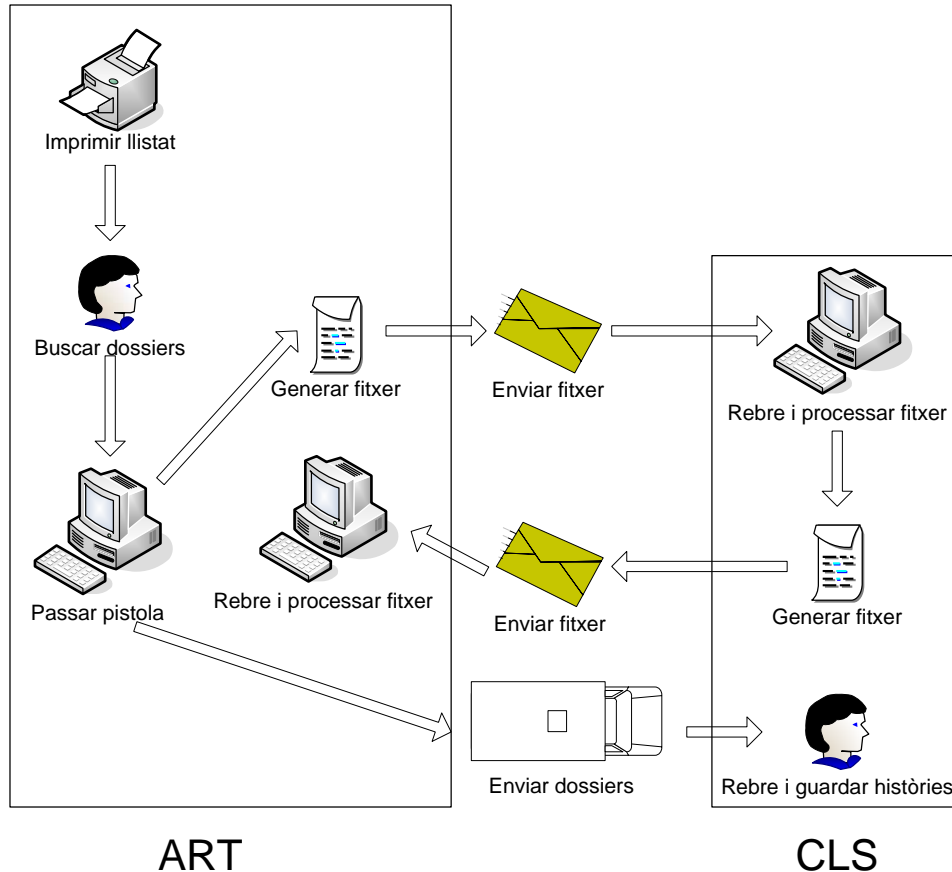


Figura 4.5. Enviament de dossiers al CLS

## 4.9 Circuit de Recuperació de dossiers

Igual que hi ha personal encarregat de prestar els dossiers, n'hi ha d'altre que és el que s'encarrega, al final del dia, d'anar a buscar les històries que ja no es necessiten, com poden ser les de les Consultes Externes que hi ha hagut al llarg del dia o les dels pacients que s'han donat d'alta, per tal de retornar-les a l'arxiu.

Així, el procediment que se segueix és, en primer lloc, anar a recollir les històries i, un cop estan a l'arxiu, llegir les corresponents etiquetes per tal que el dossier consti com a recuperat. En aquest cas és totalment indiferent el motiu pel que es va prestar el dossier, l'únic que importa és que ha retornat a l'arxiu.

Un cop, s'han marcat tots els dossiers com a recuperats, és treu el *Llistat de Prearxiu*. En aquest llistat apareixen els dossiers que s'han recuperat al llarg del dia i que tornen a tenir una petició de Consultes Externes per dintre de dos dies (és a dir, que s'hauran de preparar l'endemà). Aquestes històries es deixen a part i no es col·loquen a la seva ubicació corresponent. La resta es tornen al prestatge que toqui.

El mateix es fa amb els dossiers que han d'anar a l'arxiu AR2, per tal d'evitar portar-hi històries que s'hauran d'anar a buscar al dia següent.

Finalment, abans de portar cada dossier al seu lloc, es treu el *Llistat de dossiers recuperats*, per, mentre es van col·locant, detectar possibles errors que hi hagi hagut a l'hora de recuperar històries (els dossiers que no apareixen al llistat no s'han recuperat bé informàticament).

La Figura 4.6. mostra gràficament quin és aquest circuit.

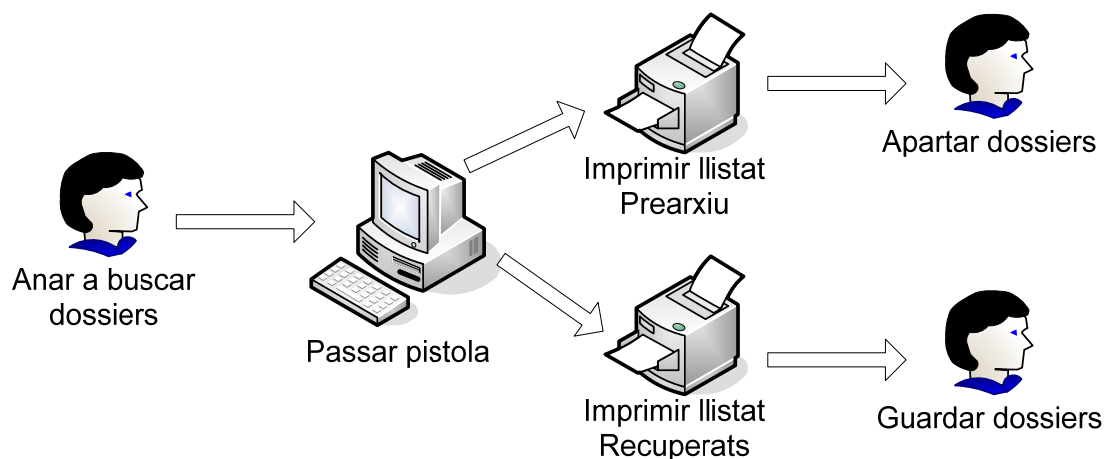


Figura 4.6. Circuit de recuperació

## **Capítol 5 Taules necessàries pel mòdul AHC**

Un cop vist com funciona l'Arxiu d'Històries Clíniques, en aquest capítol ja es comencen a explicar parts concretes que s'han creat per a la realització del mòdul AHC d'HCIS. Més concretament, en aquest capítol es veurà quines taules són necessàries per al correcte funcionament del mòdul.

### **5.1 Introducció**

La primera anotació al respecte de les taules és que, com HCIS és una aplicació que es fa servir a diferents hospitals de l'estat Espanyol, tant els noms de les taules com els de les columnes estan en castellà.

Per altre costat, és política de l'aplicació HCIS que les taules mestres tinguin codis que siguin el més explícit possibles, encara que ocupin més espai a la base de dades, per tal de fer més clar a què fan referència. D'aquesta manera, si hi ha una taula per emmagatzemar els diferents motius de sol·licitud de dossier, els codis d'aquests no seran "1, 2, 3, ...", sinó que identificaran a quin motiu referencien. Per exemple, el codi pel motiu *Consultes Externes* serà CEX, o *URG* per *Urgències*.

Parlant ja de les taules, podem diferenciar de dos tipus. Les primeres taules són les ja existents a HCIS, però que són necessàries per al correcte funcionament del mòdul AHC, com poden ser les taules *PERFIL* o *PERMISO*, entre d'altres. Per altra banda, òbviament, hi ha les taules pròpies del mòdul AHC, que s'han creat expressament per a ell. Seguint els mateixos criteris que s'han fet servir en altres mòduls, s'ha decidit que els noms de totes les taules pròpies del mòdul AHC comencin per prefix *AHC\_*.

### **5.2 Taules d'HCIS necessàries per al mòdul AHC**

En aquest punt es veuran quines de les taules que ja formen part en l'aplicació HCIS són necessàries per al correcte funcionament del mòdul AHC.

S'explicarà quina és la finalitat de la taula i quina relació té amb el mòdul AHC, però no s'entrarà a explicar la taula al detall.

#### **5.2.1 Taula CENTRO**

La CSPT és una corporació que consta de diversos Centres, estant destinat cadascun d'ells a diverses tasques. Així, per exemple, la CSPT consta del centre *HPT* (Hospital Parc Taulí), que és el centre on es fan les Urgències i les Hospitalitzacions, o del centre *CCR* (CAP Can Rull), que és un Centre d'Atenció Primària.

Un usuari, quan entra a l'aplicació HCIS, ha de triar el Centre al que vol accedir i, en funció d'aquest, tindrà accés a unes determinades funcionalitats. Cal dir que els usuaris no tenen accés a tots els centres.

El mòdul AHC només serà accessible des del centre HPT, per tant, tot usuari que vulgui accedir al mòdul haurà d'entrar a HCIS seleccionant el centre HPT.

### **5.2.2 Taula PACIENTE**

Com el seu nom indica, és en aquesta taula on es guarda la informació de tots els pacients que estan registrats a l'aplicació: nom, cognoms, data de naixement, domicili, número d'història clínica, número de targeta sanitària, etc.

La gran majoria dels panells d'HCIS (i per tant els del mòdul AHC també) mostren informació rellevant sobre el pacient del que s'estan consultant les dades, motiu pel qual és molt important aquesta taula.

Així, si s'estan consultant les dades d'un dossier, per exemple, és bàsic que aparegui informació sobre el pacient al qual pertany el dossier.

### **5.2.3 Taula USUARIO**

En aquesta taula s'emmagatzemen tots els usuaris que poden fer servir HCIS. Un professional de la Corporació que no es trobi en aquesta taula no podrà accedir a l'aplicació.

És aquí on es guarden els noms d'usuari i contrasenyes (entre d'altres dades) que fan servir els usuaris a l'hora d'accedir a l'aplicació.

És molt important que cadascú entri amb el seu codi, ja que en molt llocs s'emmagatzema l'usuari que ha realitzat l'acció.

### **5.2.4 Taula USUARIO\_CENTRO**

Com s'ha comentat amb anterioritat, no tots els usuaris tenen accés a l'aplicació per tots els centres de la CSPT. En aquesta taula es defineix a quins centres pot accedir cada usuari.

### **5.2.5 Taula PERFIL**

Entre els professionals d'un hospital hi ha molts rols diferents i cadascun té les seves pròpies tasques i funcions. No realitza la mateixa feina un metge que un administratiu o que una infermera.

En aquesta taula hi ha tots els diferents rols que pot haver-hi en un hospital, i que a HCIS es coneixen com *Perfils*. Òbviament, amb l'entrada del mòdul AHC s'ha de crear un nou perfil exclusiu per al personal de l'Arxiu d'Històries Clíniques.

Com s'ha comentat en anteriors capítols, el personal d'arxiu es connectarà a un servidor diferent que la resta dels usuaris d'HCIS. És també en aquesta taula on s'especifica a quin servidor es connectarà l'usuari, en funció de quin sigui el perfil d'aquest.

### 5.2.6 Taula USUARIO\_PERFIL

Com un usuari pot tenir més d'un perfil, existeix aquesta taula per a poder indicar a quins perfils pertany cada usuari.

L'usuari no ha de decidir amb quin perfil vol entrar a l'aplicació, sinó que té accés a totes les funcionalitats corresponents als perfils dels quals n'és membre.

Hi ha un perfil marcat per defecte per a cada usuari que és el que indica a quin servidor s'ha de connectar.

### 5.2.7 Taula PRIVILEGIO

Els privilegis indiquen a quines parts de l'aplicació es pot accedir. Per exemple, hi ha un privilegi per accedir a la *Gestió de Dossiers*, o a la *Consulta de Pacients Hospitalitzats*.

### 5.2.8 Taula PERFIL\_PRIVILEGIO

Els privilegis s'associen als perfils, i és en aquesta taula on s'especifiquen les relacions existents. Així, per posar un exemple, el perfil ARXIU tindrà el privilegi per a accedir a la *Gestió de Dossiers*, però no a la *Consulta de Pacients Hospitalitzats*.

### 5.2.9 Taula PERMISO

A part dels privilegis, hi ha els permisos. Així com els privilegis indiquen a quines parts de l'aplicació es pot accedir, els permisos indiquen quines accions es poden executar. Així, per exemple, dintre de la *Gestió de Dossiers*, hi ha un permís per a *Crear Dossier*, un altre per a *Modificar Dossier*, o per *Eliminar Dossier*.

### 5.2.10 Taula PERMISO\_AMBITO

És en aquesta taula on s'indica quines accions pot realitzar cada usuari. Un permís es pot donar a nivell de perfil o d'usuari. Per exemple, tot el personal d'Arxiu, dintre de la *Gestió de Dossiers*, té permís per a *Consultar Dades* d'un dossier. Per tant, hi haurà un registre a aquesta taula, marcant que l'àmbit és PERFIL, indicant que

ARXIU té permís per a Consultar Dades d'un dossier. En canvi, només uns usuaris específics poden *Eliminar Dossiers*. D'aquesta manera, en aquesta taula s'indicarà, marcant que l'àmbit és USUARI, quins professionals poden realitzar aquesta acció.

### **5.2.11 Taula EPISODIO**

Cada cop que un pacient va a l'hospital se li obre un episodi, que es tanca una vegada el pacient l'abandona pel motiu que sigui. En aquesta taula es té informació de la data i hora d'arribada del pacient, del motiu, de la data i hora de tancament de l'episodi (si està tancat), etc. Depenent del motiu de la visita hi ha diferents tipus d'episodis.

Així, HCIS té episodis d'*Urgències* si el pacient ha entrat a urgències, d'*Hospitalització* si el pacient ha ingressat, i *Ambulatoris*, si ha acudit a una visita de consultes externes. Com les intervencions quirúrgiques acostumen a necessitar un ingrés, s'associen a episodis d'Hospitalització. Si són intervencions que no precisen ingrés, com pot ser una intervenció de cataractes, s'associen a episodis Ambulatoris.

### **5.2.12 Taula ADM\_MOVIMIENTO**

Quan un pacient està ingressat, tots els moviments que té s'emmagatzemen en aquesta taula. Així, si es vol saber a quina habitació, llit o planta està ingressat un pacient, s'ha d'accedir a la taula ADM\_MOVIMIENTO. Aquesta informació pot resultar molt útil al personal d'arxiu per quan han de portar dossiers de pacients que ja es troben ingressats.

### **5.2.13 Taula URG\_MOVIMIENTO**

Aquesta taula és similar a l'anterior, però pels pacients que es troben a urgències.

### **5.2.14 Taula CEX\_CITA**

La taula CEX\_CITA és on s'emmagatzemen totes les visites de Consultes Externes. Aquesta taula és molt important ja que totes les peticions de dossier de consultes externes PROGRAMADES es realitzen, en funció de l'agenda associada a la consulta externa, quan es programa la visita i s'insereix un nou registre a CEX\_CITA.

### **5.2.15 Taula CEX\_AGENDA**

És en aquesta taula on s'emmagatzemen les diferents agendes de Consultes Externes. Associat a cada agenda hi ha un camp que indica si, per a les visites de l'agenda, s'ha de fer una petició de dossier a l'Arxiu. Així, si aquest camp val 'S', en el moment que es programi una visita per a aquesta agenda, automàticament es crearà una



petició de dossier per al dia i hora de la visita. Si aquest camp val 'N' i es desitja el dossier, s'ha de fer manualment una petició espontània.

### **5.2.16 Taula QUI\_PARTE**

Aquesta és la taula on hi ha tota la informació de les intervencions quirúrgiques. Al personal d'arxiu li és útil per a les peticions de dossiers que es fan amb motiu de una operació.

### **5.2.17 Taula MENSAJE**

A HCIS tots els textos que apareixen per pantalla estan emmagatzemats en aquesta taula, tant en català com en castellà.

La taula MENSAJE té un camp CODI, per a identificar el missatge, un camp IDIOMA, que indica en quin idioma està el missatge, i un camp TEXTO, que és el missatge en sí. Així, en comptes d'escriure el missatge, s'especifica el codi d'aquest i HCIS, en funció de l'idioma associat a l'usuari, el mostra en una llengua o una altra.

### **5.2.18 Taula PARAMETRO**

Finalment, HCIS té moltes de les seves funcionalitats parametritzades. Així en funció del valor d'un paràmetre realitza una acció o una altra. D'aquesta manera es pot parametritzar, per exemple, que quan es creï un dossier s'imprimeixi automàticament una etiqueta de codi de barres per enganxar-la-hi.

## **5.3 Taules pròpies del mòdul AHC**

En aquest punt s'expliquen quines taules s'han creat per a guardar tota la informació necessària per a poder integrar el mòdul AHC dintre de l'aplicació HCIS.

També s'indica quina és la finalitat de la taula, així com quins camps la formen i marcant en negreta els camps que formen la clau primària.

A més, com hi ha moltes taules que conformen el mòdul AHC d'HCIS i, òbviament, aquestes estan relacionades les unes amb les altres, també es mostraran aquestes relacions, per tractar d'aclarir com estan lligades unes taules amb les altres. Per no carregar tant les imatges, només s'introdueixen els camps que són claus primàries de les taules amb la que es relaciona la taula que s'estigui explicant. No es mostrarà la relació de totes les taules entre elles, ja que l'esquema no seria gens clar.

### 5.3.1 Taula AHC\_ARCHIVO

Aquesta és la taula mestra que conté els diferents arxius existents a l'hospital. És a dir, *ART*, *AR2* i *CLS*.



Figura 5.1. Taula AHC\_ARCHIVO

### 5.3.2 Taula AHC\_CAMBIOESTADO

Per tal de portar un històric de tots els moviments que ha tingut un dossier, s'ha creat aquesta taula. Així, per cada moviment que tingui un dossier, s'inserirà un registre a AHC\_CAMBIOESTADO. És a dir, aquesta taula és un històric de cada dossier.

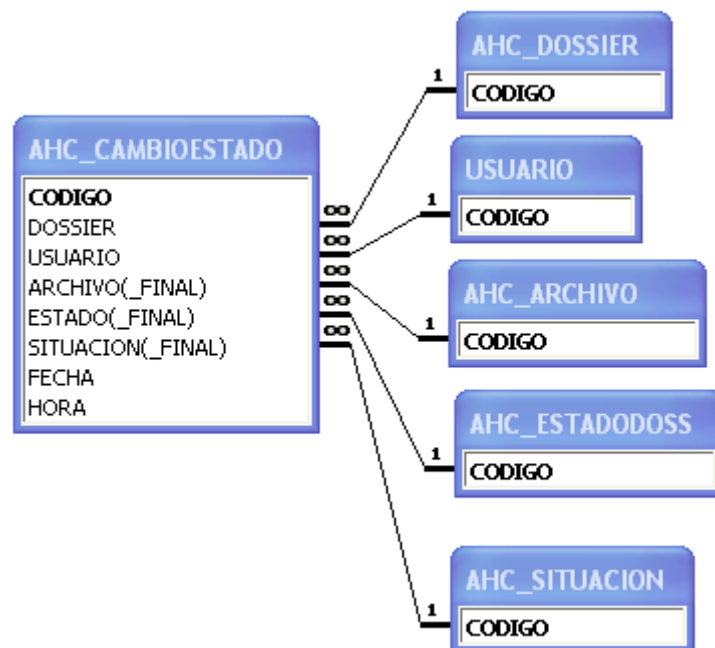


Figura 5.2. Taula AHC\_CAMBIOESTADO i les seves relacions

### 5.3.3 Taula AHC\_CATEGORIA

Per tal que el personal d'arxiu sàpiga quines peticions s'han de servir primer hi ha diferents categories. Aquesta taula és el mestre de categories. Les que es fan servir a la CSPT són *Normal*, *Preferent* i *Urgent*.



Figura 5.3. Taula AHC\_CATEGORIA

### 5.3.4 Taula AHC\_DOSSCENTRO

Aquesta taula és la que conté quins tipus de dossier hi ha a cada centre. A la CSPT només hi ha un tipus de dossier, que està disponible a tots els centres de la Corporació.

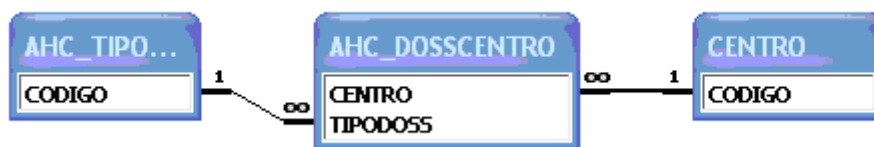


Figura 5.4. Taula AHC\_DOSSCENTRO i les seves relacions

### 5.3.5 Taula AHC\_DOSSIER

Aquesta és la principal taula del mòdul AHC, ja que és on es guarda la informació actual de cadascun dels dossiers registrats a l'aplicació.

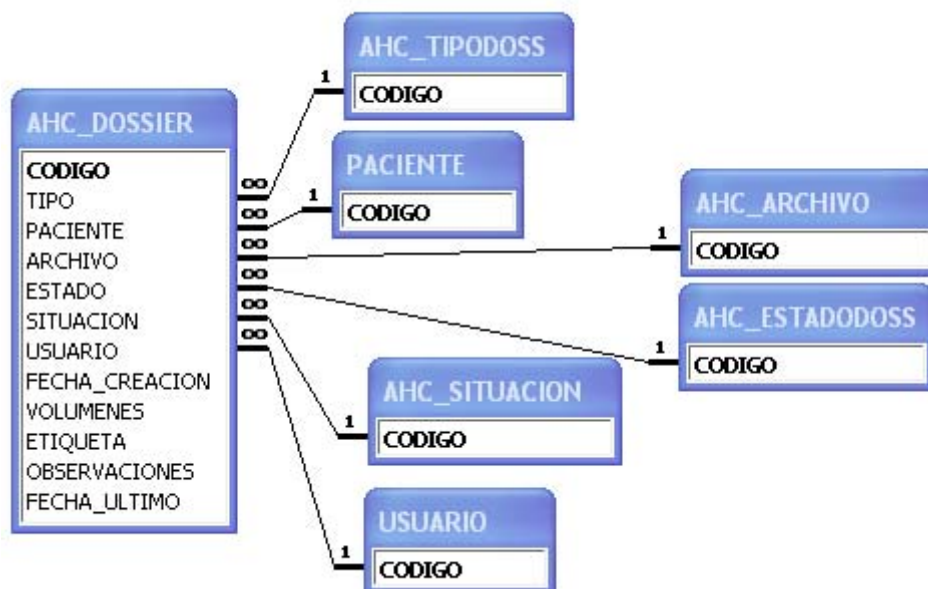


Figura 5.5. Taula AHC\_DOSSIER i les seves relacions

### 5.3.6 Taula AHC\_ERRORPREP

Aquesta és la taula mestra dels possibles errors de preparació. Quan es preparen les peticions de dossiers, poden quedar-se en dos estats, o bé PREPOK si el dossier es troba a l'arxiu, o bé PREPERERROR, si el dossier és exterior. En cas que el dossier sigui exterior, s'ha d'indicar el motiu, que pot ser que està prestat, o que està al CLS, o fins i tot que s'ha extraviat.

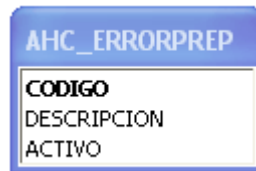


Figura 5.6. Taula AHC\_ERRORPREP

### 5.3.7 Taula AHC\_ESTADODOSS

Aquesta taula és el mestre de possibles estats d'un dossier. Un dossier pot estar en estat *Disponible*, *Prestat*, *Extraviat*, etc

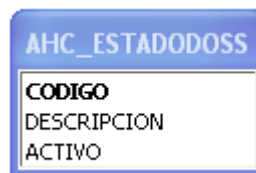


Figura 5.7. Taula AHC\_ESTADODOSS

### 5.3.8 Taula AHC\_FUSION

Quan un pacient va a l'hospital, el primer que es fa és comprovar si ja havia assistit amb anterioritat i, per tant, ja té número d'història clínica. De vegades, però, s'obren noves històries clíniques a pacients ja existents, motiu pel que, de vegades, hi ha pacients amb dues històries clíniques. Quan es detecten aquests casos, és fusionen les dues històries en una. Aquesta tasca no es fa des del mòdul AHC, però sí que rep una notificació per saber que hi ha hagut dues històries que s'han fusionat en una de sola. Aquesta és la taula encarregada d'emmagatzemar aquesta informació.

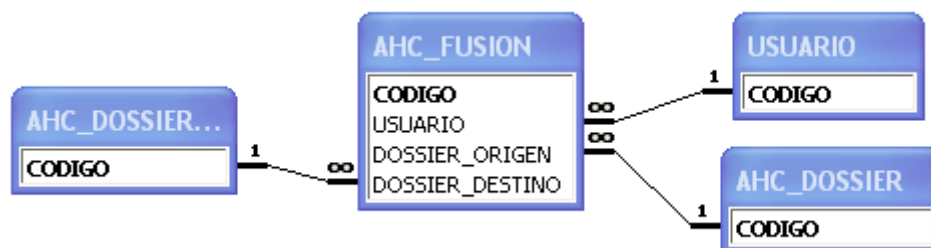


Figura 5.8. Taula AHC\_FUSION i les seves relacions

### 5.3.9 Taula AHC\_HPRESTAMO

Aquesta taula és un històric de tots els préstecs que s'han fet de cada dossier. Si un dossier està prestat en l'actualitat, també es troba en aquesta taula, però té la data de devolució en blanc.

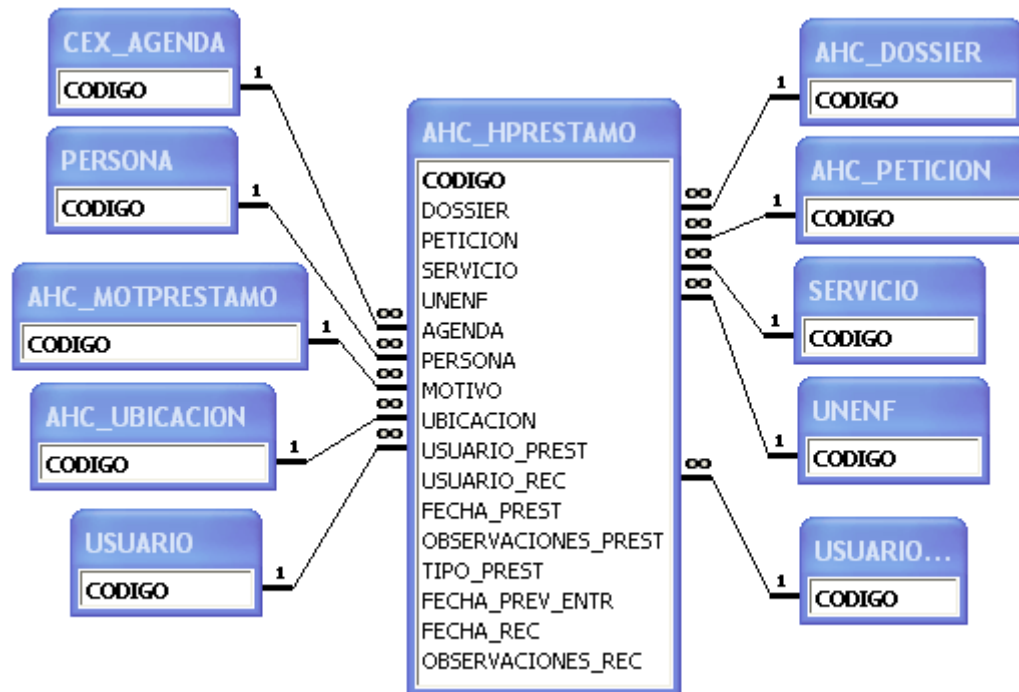


Figura 5.9. Taula AHC\_HPRESTAMO i les seves relacions

### 5.3.10 Taula AHC\_MOTPRESTAMO

Aquesta és la taula que emmagatzema els diferents motius per els que es pot sol·licitar i prestar un dossier, com poden ser *Consultes Externes*, *Urgències*, *Estudis*, ...

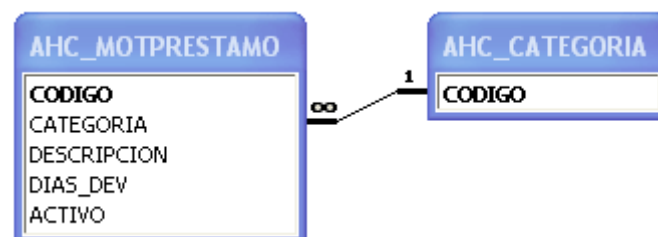


Figura 5.10. Taula AHC\_MOTPRESTAMO i les seves relacions

### 5.3.11 Taula AHC\_MOTUBI

Depenent de quin sigui el motiu de sol·licitud d'un dossier, hi ha una sèrie d'ubicacions on es pot portar aquesta història clínica. Per exemple, si el motiu és Consultes Externes, el dossier es pot portar a qualsevol de les diferents sales d'espera que hi ha a la CSPT (i que estan definides a la taula AHC\_UBICACION), però no es

pot portar a una sala d'urgències (que també esta definida a AHC\_UBICACION). És a dir, hi ha una relació de N a N entre les taules AHC\_MOTPRESTAMO i AHC\_UBICACION. Aquesta taula és l'encarregada de relacionar cada motiu de sol·licitud (o préstec) amb les ubicacions on poden anar en funció del motiu.



Figura 5.11. Taula AHC\_MOT\_UBI i les seves relacions

### 5.3.12 Taula AHC\_PETICION

En aquesta taula s'emmagatzemen totes les peticions de dossiers existents, tant les passades, com les que estan prestades en l'actualitat i les que s'han fet a futur.

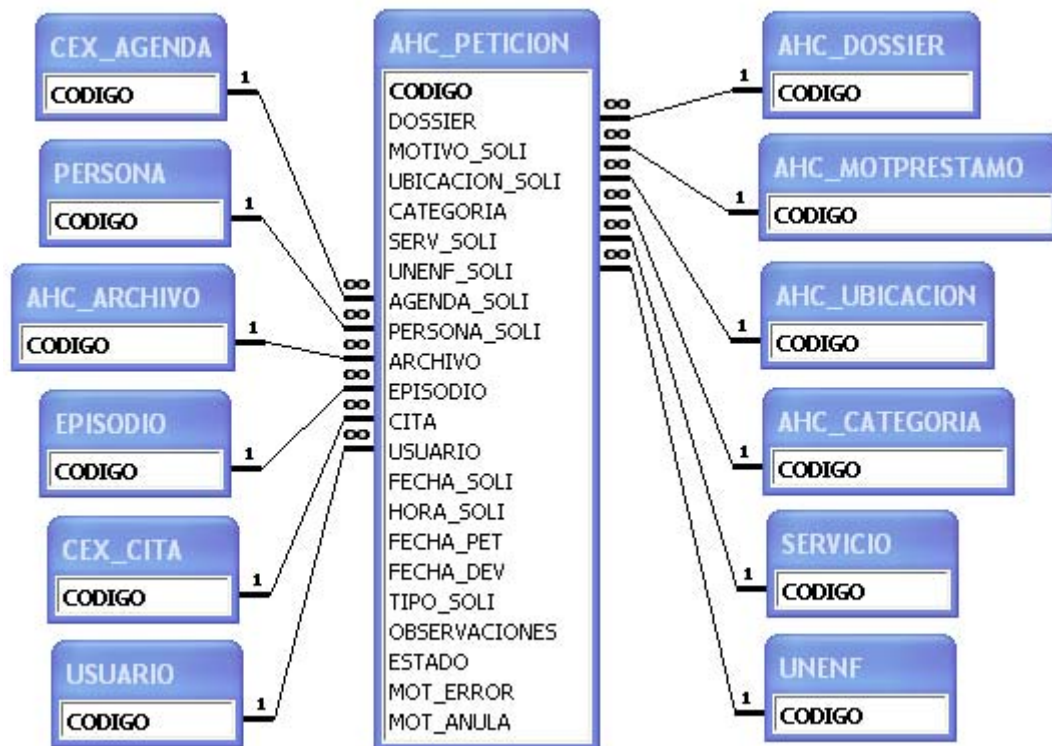


Figura 5.12. Taula AHC\_PETICION i les seves relacions

### 5.3.13 Taula AHC\_PRESTAMO

En aquesta taula trobem els dossiers que es troben prestats en l'actualitat.

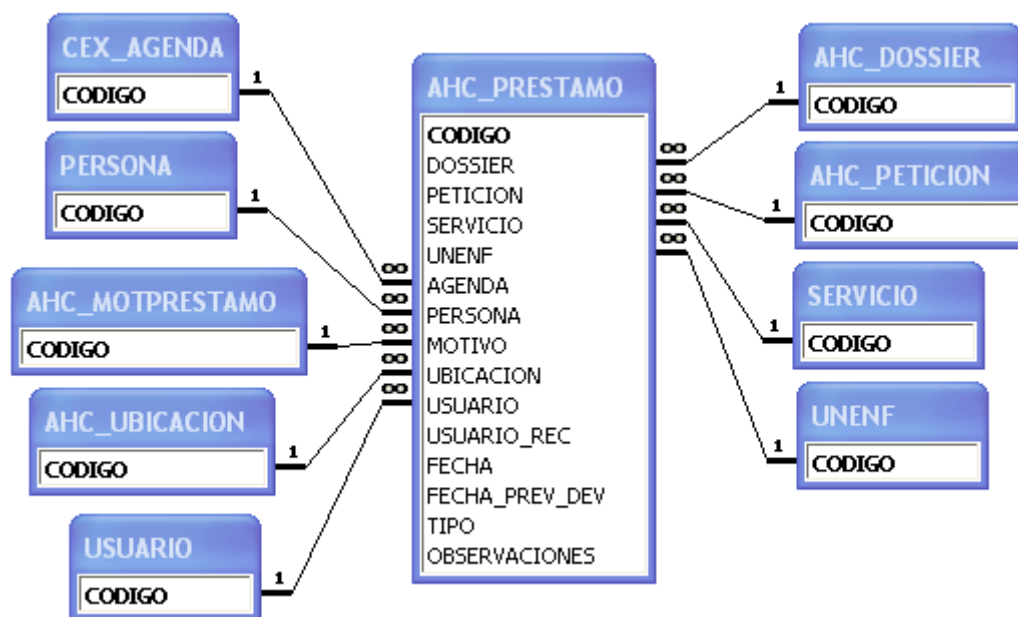


Figura 5.13. Taula AHC\_PRESTAMO i les seves relacions

### 5.3.14 Taula AHC\_SITUACION

Aquesta és la taula mestra de les possibles situacions en què es pot trobar un dossier. Aquestes són *Actiu*, *Passiu*, *Semipassiu* i *Èxitus*. Els dossiers dels pacients morts tenen situació Èxitus. De la resta, els que fa poc temps que han tingut moviments estan en situació Actiu, mentre que les situacions Passiu i Semipassiu es reserven per als dossiers que fa temps que no tenen moviments (Passiu vol dir que fa més temps que no té moviments). Els criteris per a posar un dossier en una situació o una altra els defineix l'usuari, ja que no hi ha cap automatisme que canviï les situacions. Per exemple, a la CSPT s'ha decidit que els dossier que faci menys de dos anys que han tingut moviments estaran en situació Actiu. Els que fa entre dos i quatre anys que no tenen moviments estan en situació Semipassiu. I els que fa més de quatre anys que no tenen moviments estan en situació Passiu.

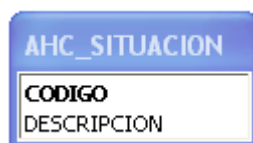


Figura 5.14. Taula AHC\_SITUACION

### 5.3.15 Taula AHC\_TIPODOSS

En aquesta taula s'indiquen quins tipus de dossier hi ha.

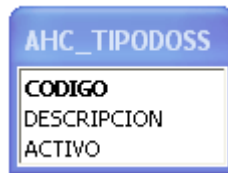


Figura 5.15. Taula AHC\_TIPODOSS

### 5.3.16 Taula AHC\_UBICACION

Aquesta és la taula mestra que especifica a quines ubicacions es pot portar una història.



Figura 5.16. Taula AHC\_UBICACION

### 5.3.17 Taula AHC\_VOLUMEN

Quan un dossier és tan gran que tota la informació no cap al mateix arxivador, s'ha d'agafar-ne un de nou, però segueix sent el mateix dossier, tot i que és un nou volum d'aquest. En aquesta taula només s'emmagatzemen els volums d'aquells dossiers que en tenen més d'un.

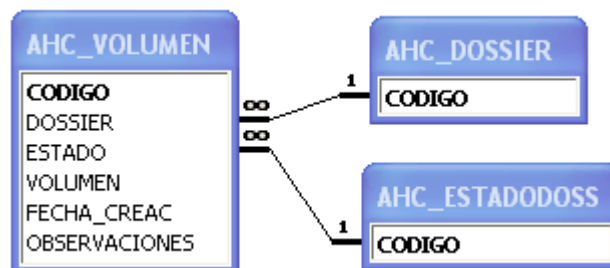


Figura 5.17. Taula AHC\_VOLUMEN i les seves relacions



## 5.4 Conclusions

En aquest apartat s'han vist totes les taules pròpies del mòdul AHC i d'altres pròpies d'HCIS que també són importants per al mòdul per tal de poder obtenir altra informació, com poden ser les dades d'un pacient, per exemple.

Com s'ha pogut veure, l'elevat nombre de taules que hi intervenen fan que el mòdul AHC sigui força complex. És per aquest motiu que hi ha dades duplicades en algunes taules, per tal d'aconseguir creuar quantes menys taules millor. Degut al gran volum de dades amb les que ha de treballar el mòdul AHC, aquest fet també afavorirà a l'hora d'obtenir determinada informació. Cal notar que aquest aspecte ha vingut marcat des de Producte.

## **Capítol 6 Implementació del mòdul AHC**

A l'hora de programar qualsevol aplicació informàtica és convenient seguir unes petites pautes en l'estructura del codi i en l'estil de programació, per tal de fer aquest més entenedor a altres programadors que puguin mirar la codificació de l'aplicació en un futur. D'aquesta manera, al ja estar familiaritzats tant amb l'estructuració com amb l'estil de programar, només s'hauran d'entretenir en entendre la funcionalitat de l'aplicació. Aquest aspecte encara és més important quan es tracta d'una aplicació en la que hi treballa molta gent, ja que la possibilitat de que algú que no sigui el programador inicial del mòdul l'hagi de modificar és força elevada.

En aquest capítol s'explicaran quines han estat les pautes que s'han seguit i per quin motiu s'han escollit.

A més, en aquest apartat també s'explicaran tant les classes Java com els arxius jsp (JavaServer Pages) que conformen el mòdul AHC.

### **6.1 Introducció**

En primer lloc es farà una breu explicació sobre el llenguatge de programació que fa servir HCIS i que, per tant, és el mateix en el que s'ha programat el mòdul AHC. Com ja s'ha comentat amb anterioritat, es fa servir la plataforma *J2EE* (Java Platform Enterprise Edition), que és una plataforma per a desenvolupar aplicacions Java des del costat del servidor.

#### **6.1.1 Java**

Java és un llenguatge de programació orientat a objectes que presenta una portabilitat total entre màquines. Això vol dir que Java està dissenyat per evitar els problemes que es poden presentar durant l'execució de codi en entorns distribuïts, on es poden trobar màquines amb sistemes operatius diferents.

Java és un llenguatge molt simple, totalment orientat a objectes, que permet la programació concurrent, cosa que li proporciona unes altes prestacions en aplicacions que necessiten fer anar activitats en paral·lel. A més, és un programa molt segur, fins i tot a l'hora de baixar aplicacions d'Internet degut al fet que el sistema run-time de Java incorpora protecció per virus i d'altres problemes.

Aquest, doncs, és un llenguatge ideal per a desenvolupar aplicacions d'usuari segures, distribuïdes i basades en xarxa.

Aquest llenguatge ofereix una gran facilitat per programar, ja que permet la reutilització de codi i la seguretat d'accés a memòria, entre d'altres. D'aquesta manera es genera codi depurat fàcilment.

A més, Java incorpora el Garbage Collector, que ens estalvia la reserva explícita de memòria pel fet que aquest element elimina la memòria que s'ha creat en una funció quan en surt d'aquesta. També utilitza les referències de tipus segur i el llançament d'excepcions per tal d'evitar els errors més comuns en els llenguatges.

Finalment, cal notar que tots els programes Java estan construïts a partir de classes. A partir de cada una d'aquestes classes es generarà un fitxer de codi bytecode. A partir de la definició d'una classe es pot crear qualsevol tipus d'objecte.

### 6.1.2 JavaServer Pages (JSP)

JavaServer Pages és una tecnologia Java que permet generar contingut dinàmic per webs, que permeten utilitzar codi Java mitjançant *scripts*. JSP es pot considerar una manera alternativa i simplificada de construir *servlets* (programes que s'executen en servidors web, dissenyats per oferir contingut dinàmic).

El funcionament general d'aquesta tecnologia és que el Servidor d'Aplicacions interpreta el codi contingut a la pàgina JSP per construir el codi Java del servlet a generar. El servlet serà qui generi el document que es presentarà a la pantalla del navegador de l'usuari.

Els JSPs són en realitat servlets: la primera vegada que s'invoca un JSP es compila a un programa Java, creant-se una classe que es comença a executar al servidor com un servlet. La principal diferència que hi ha entre els JSPs i els servlets és des de l'enfocament de la programació, ja que un JSP és un programa Web amb unes etiquetes especials y codi Java incrustat, mentre que un servlet és codi Java pur que rep peticions i genera una pàgina web a partir d'elles.

## 6.2 Estil de programació

Un cop explicada breument quina tecnologia es fa servir a HCIS, a continuació s'explicarà quines pautes s'han seguit a l'hora de programar el mòdul AHC que, com es de suposar, seran les mateixes que es fan servir a la resta de l'aplicació.

Com a estil de programació, HCIS segueix el criteri més utilitzat entre la comunitat de programadors i que és el que segueix les pautes que s'expliquen en aquest apartat. D'aquesta manera s'aconseguirà que el codi sigui més comprensible i clar.

Al ser Java un llenguatge que diferencia entre les majúscules i les minúscules, s'utilitzaran aquestes per tal de distingir-ne el tipus d'element que es tracta, ja sigui variable, classe, mètode, etc.

## 6.2.1 Variables i mètodes

Tant les variables com els mètodes comencen sempre per minúscula: **codigo**, **inicializar**( ). Es diferencien entre ells pel fet que els mètodes van sempre seguits per parèntesis.

Si el nom de l'identificador està format per més d'una paraula, la primera lletra de la següent paraula és majúscula: **panellPrincipal**, **getCodigo** ( ). Hi ha però una excepció, i és en els constructors de les classes que, com tenen el mateix nom que la classe, i el nom d'aquesta comença en majúscula (com es veurà al següent punt), el mètode també començarà amb majúscula: **AhcArchivo**( ).

## 6.2.2 Classes

Les classes que ja proporciona Java comencen sempre per majúscula: **String**, **Servlet**, ... Per tant, totes les classes que formen el programa tindran la seva primera lletra majúscula i, d'aquesta manera, s'identificaran més ràpidament. En el cas que el nom de l'identificador tingui més d'una paraula, també s'usarà la majúscula com a separador: **AhcArchivo**.

## 6.3 Estructura del codi

Igual que en el cas anterior, el criteri que s'ha seguit a l'hora d'estructurar el codi ha estat el que s'ha fet servir al llarg de tots els anys de carrera. A continuació s'explica quin és aquest criteri.

### 6.3.1 Formats dels blocs de codi

Quan un bloc de codi consta de més d'una línia, aquest s'ha d'escriure entre claus "{ }" per tal de marcar-ne l'inici i el final. Així, la clau que obre el bloc de codi s'escriu a la mateixa línia de la instrucció a la qual pertany, mentre que la clau encarregada de tancar-lo es posa a una línia sola al final del bloc.

La codificació de les instruccions del bloc se situen uns espais més a la dreta de la instrucció inicial. D'aquesta manera és molt més fàcil localitzar el principi i el final del bloc.

```
for(int fila=0;fila<fil;fila++) {  
    for(int colu=0;colu<col;colu++) {  
        array[fila][colu] = new Vector();  
        lista.add(array [fila][colu]);  
    }  
}
```

En canvi, quan el bloc està format únicament per una línia, aquesta s'escriu a continuació de la instrucció, sempre i quan la línia de bloc no sigui molt llarga.

**if(error) mostrarError(valores-1);**

Per altra banda, les instruccions “try/catch” s'escriuen amb el terme “catch” a continuació de la clau que tanca el terme “try”.

```
try {  
    codi_try;  
} catch() {  
    codi_catch;  
}
```

Les construccions “if/else” tenen el mateix format que les “try/catch”, sempre i quan els blocs de codi que els acompanyen estiguin formats per més d'una lletra.

Pel que fa a les pàgines JSP, per intercalar el codi Java dintre el codi HTML, es fan servir les marques `<% i %>`, per a indicar l'inici i el fi del bloc. Si el bloc de codi Java ocupa una línia, s'obre i es tanca a la mateixa línia. Sinó, s'obre el bloc, es comença amb la part del codi a la següent línia i, un cop finalitzat el bloc, es tanca a una nova línia.

```
<%codi Java que només ocupa una línia%>  
<%  
codi Java  
que ocupa  
més d'una línia  
%>
```

### 6.3.2 Comentaris

Per tal de fer el codi encara més entenedor, als programes se'ls hi poden afegir comentaris per tal d'aclarir qualsevol dubte que hi pugui haver, com poden ser explicar la funcionalitat d'un mètode o la utilització d'una determinada variable. Aquests comentaris són ignorats a l'hora de compilar i executar el programa.

Hi ha dos tipus de comentaris: els que ocupen una sola línia i els que ocupen més d'una línia. Els primers s'identifiquen perquè comencen amb l'identificador `//`, mentre els que ocupen més d'una línia comencen amb `/*` i es tanquen amb `*/`.

**//això és un comentari que ocupa una línia.**  
**/\*això és un comentari que ocupa més d'una línia. Tot el que s'escriu dintre dels identificadors d'inici i fi de comentari s'ignora durant la compilació i l'execució del programa.\*/**

Si el comentari es troba a un fitxer JSP, a la part HTML, el comentari s'obrirà amb `<!--` i es tancarà amb `-->`.

A l'hora de posar els comentaris en el codi, s'han seguit les següents pautes:

1. Si el comentari fa referència a una sola línia del codi, aquest es posa al final de la mateixa.

```
private String codigo;//código del objeto AhcArchivo
```

2. Si el comentari fa referència a tot un bloc de codi, aquest se situa abans del començament del bloc.

```
//función que retorna el atributo codigo del objeto AhcArchivo
```

```
public int getCodigo( ) {  
    return this.codigo;  
}
```

## **6.4 Clases per interactuar amb la base de dades**

En aquest apartat es descriuran les diferents classes que s'ha hagut de crear perquè el mòdul AHC pugui treballar amb les dades que hi ha emmagatzemades a la base de dades. S'explicarà quina és la seva finalitat i quines són les seves principals funcions.

### **6.4.1 El concepte de Datamanager**

Abans de tot, s'ha d'explicar que HCIS implementa el concepte de *Datamanager*, d'ara en endavant DM. Aquest concepte consisteix en carregar en memòria quanta més informació millor, per tal de minimitzar els accessos a base de dades, ja que és més ràpid accedir a memòria que no pas fer una consulta a base de dades.

HCIS treballa amb dos tipus diferents de DMs, podent-se definir per a cada DM de quin tipus ha de ser. El primer és el que, quan s'arrenca l'aplicació, carrega en memòria tota la informació de la taula corresponent i no farà cap accés a base de dades per a la consulta a memòria. És el tipus de DM que es fa servir per a les taules mestres, ja que la informació que contenen aquestes taules acostuma a ser força estàtica i a canviar poc, com pot ser la taula AHC\_ARCHIVO.

Per altra banda, hi ha el tipus de DM que es fa servir per a les taules que tenen dades més canviants, com pot ser la taula AHC\_DOSSIER. En arrencar-se l'aplicació, aquest DM està buit. Quan s'ha d'accedir a un registre d'aquesta taula, en primer lloc, es va a buscar al DM. Si el troba, es treballa amb la informació que conté el DM. Sinó, es va a buscar a base de dades i s'insereix al DM. D'aquesta manera, si posteriorment s'ha de tornar a accedir a aquest registre, aquest ja es trobarà al DM i s'estalviarà un accés a base de dades. A aquests tipus de DMs se'ls especifica un màxim nombre d'elements en memòria, ja que sinó es podria arribar a carregar tota la taula, podent deixar a l'aplicació sense recursos suficients per al seu correcte funcionament. Quan el

DM arriba a la seva mida màxima, si ha d'inserir un nou element en memòria, elimina el que fa més temps que no es fa servir.

Cal remarcar que si, des de l'aplicació, s'insereix un nou registre o es modifica o elimina un d'existent, aquest s'afegeix o es canvia o elimina tant a la taula com al DM corresponents.

En canvi, si la inserció, modificació o eliminació es fa directament a la base de dades, pot ser que l'aplicació no se n'assabenti. Si és un DM del primer tipus no s'assabenta perquè està carregat totalment a memòria i no s'accedeix mai a base de dades a l'hora de fer consultes. Si és del segon tipus, pot ser que sigui d'un registre que ja s'hagi consultat i que, per tant, ja estigui carregat al DM corresponent. En aquest cas, l'aplicació tampoc sabria que s'ha produït un canvi.

Per aquest motiu, HCIS consta amb un punt manual a l'aplicació de *Refresc de Datamanagers*. Quan es refresca un DM, si és del primer tipus, es torna a carregar tota la informació en memòria, mentre que si és del segon, el que fa és buidar el DM. Així, quan s'hagi d'accedir a un registre, si o bé ja s'haurà tornat a carregar el DM, o bé s'haurà buidat i, per tant, s'haurà d'accedir a base de dades, obtenint les dades actualitzades.

## 6.4.2 Classes per treballar amb les dades

Per tal de treballar amb les dades emmagatzemades, HCIS consta amb dos tipus de classes.

El primer tipus de classe és la que deriva de la classe *DatamanagerCache.java*, són les que gestionen totes les operacions del DM i els accessos a la base de dades. Les classes d'aquest tipus tenen el nom de la taula, acabat en DM, per exemple *AhcArchivoDM.java*, i són les encarregades de carregar la informació corresponent al DM i fer les cerques de dades, així com les insercions, modificacions i eliminacions a la base de dades.

Totes les classes d'aquest tipus implementen les següents funcions:

### · ***NomTaulaDM()***

És el constructor de la classe. Si el DM és del primer tipus (dels que carreguen les dades en memòria), l'únic que fa és inicialitzar la variable *sentencia*. Aquesta variable indica quina consulta a base de dades s'ha de llançar per carregar el DM corresponent.

Si el DM és del segon tipus, és a dir dels que es van carregant conforme es treballa amb les dades, és un constructor buit.

### · ***cargaDatos()***

És la funció que s'encarrega de carregar les dades al DM de la taula, si el DM és d'una taula mestra. S'executa al arrencar l'aplicació i quan es fa el refresc manual de DM. Aquesta funció llança la sentència especificada al constructor contra base de dades, esborra les dades que hi havia memòria i carrega les que ha retornat la consulta.

En cas que el DM sigui del segon tipus, l'únic que fa aquesta funció és buidar aquest DM.

· **getObject(Vector fila)**

Quan es llança una consulta contra base de dades, cada registre es retorna en forma de Vector, on cada posició del vector correspon a un camp de la taula. Aquesta funció s'encarrega de transformar el Vector en un objecte.

Retorna l'objecte que s'ha creat amb les dades del vector.

· **update(NomTaula objectUpdate)**

Rep l'objecte ja amb les dades correctes i s'encarrega de llançar una consulta d'actualització a la base de dades. Els paràmetres són els atributs de l'objecte rebut. El camp clau serà el que es farà servir per a indicar quin objecte s'ha d'actualitzar (clàusula WHERE del update).

· **insert(NomTaula objectInsert)**

Rep l'objecte que s'ha d'inserir i llança la consulta d'inserció. Els paràmetres són els atributs de l'objecte *objectInsert*.

· **delete(NomTaula objectDelete)**

Rep l'objecte que s'ha d'eliminar i llança la consulta d'eliminació. Té un sol paràmetre, que és el camp clau de l'objecte *objectDelete*, i que es farà servir per a indicar quin objecte s'ha d'eliminar (clàusula WHERE del delete).

· **findByClave(String codigo)**

Rep una clau primària i retorna l'objecte que la té.

· **getDatos(Map parametros)**

Rep un conjunt de paràmetres i retorna un Vector amb tots els objectes que compleixen les condicions indicades als paràmetres.

Per altra banda, hi ha les classes que deriven de *DataItem.java* i que són les que tracten els registres com objectes, per exemple *AhcArchivo.java*. Tenen un atribut per a cada columna de la taula, amb els mètodes corresponents *get()* i *set()*, per a poder obtenir i establir el valor de cadascun.

A més, aquestes classes tenen un constructor buit, que crea un objecte amb tots els atributs en blanc, i un constructor que rep els atributs necessaris per a crear l'objecte amb dades. També implementen totes les funcions pròpies (les que s'han indicat més amunt) que tingui la classe DM associada (per exemple la classe *AhcArchivo.java* implementarà les funcions pròpies de la classe *AhcArchivoDM.java*). Finalment, també implementen tots els mètodes següents:

· **getDatamanagerName()**

Retorna el nom del DM amb la seva ruta.

· **getDatamanager()**

Retorna una instància d'un objecte del tipus del nom que retorna la funció *getDatamanagerName()*. Cal remarcar que les classes que fa les consultes a base de



dades o a memòria són les del tipus *NomTaulaDM.java*, però que des de la resta del codi es treballa amb objectes dels tipus de les classes *NomTaula.java*. Així, si es volen obtenir tots els objectes *AhcArchivo*, es cridarà a la funció *getAll()* de la classe *AhcArchivo.java*, que a la seva vegada cridarà a la funció *getAll()* de la classe *AhcArchivoDM.java* de la següent manera: **`getDatamanagerName().getAll()`**;

- **`findByClave(String codigo)`**

Crida a la funció *findByClave(String codigo)* de *NomTaulaDM.java* i en retorna el resultat obtingut.

- **`getParametros()`**

Retorna una llista on cada element és un camp de l'objecte per al qual es crida la funció, fent servir les funcions '*get*' corresponents.

- **`getMapParametros()`**

Retorna un objecte Hashmap (descripció + valor) on cada element és un camp de l'objecte per al qual es crida la funció. La descripció de l'element és el nom del camp i el valor el que retorni la funció '*get*' corresponent.

- **`getDatos(Map parametros)`**

Crida a la funció *getDatos(Map parametros)* de *NomTaulaDM.java* i en retorna el resultat obtingut.

- **`clonar()`**

Fent servir els mètodes '*get*', crea un objecte exactament igual que el que està invocant el mètode.

- **`equals(Object o)`**

Retorna *true*, si l'objecte que es passa per paràmetre és exactament igual que el que ha cridat el mètode. Retorna *false* en cas contrari.

- **`salvar()`**

La classe *DataItem.java*, que és de la que deriven aquests tipus de classes, té tres variables booleanes, *\_nuevo*, *\_modificado* i *\_eliminado*. Només una d'aquestes tres variables pot valer *true*. En funció de l'atribut que estigui activat, es cridarà a la funció *insert()*, *update()* o *delete()* de la classe *NomTaulaDM.java*. Quan es crida qualsevol mètode *set*, la variable *\_modificado* es posa a *true*. Quan es crea un nou objecte, al final de tot s'ha de posar la variable *\_nuevo* a *true* i la *\_modificado* a *false*, ja que, si s'han fet servir mètodes '*set*' per a establir els valors, valdria *true* també.

- **`eliminar()`**

Posa la variable *\_eliminado* a *true*, les variables *\_modificado* i *\_nuevo* a *false* i crida a la funció *salvar()*.

A continuació s'expliquen les classes d'aquests tipus que formen el mòdul AHC.

#### 6.4.2.1 Classe AlmacenSqlHCIS.java

HCIS consta amb una classe on hi ha les consultes que s'han de llançar contra base de dades. Cada consulta és un mètode. Perquè s'executi la consulta desitjada només s'ha d'executar la funció interna d'HCIS *ejecutarSentencia()*, passant-li el nom de la sentència que es vol executar i una llista amb els paràmetres necessaris perquè s'executi la consulta.

#### 6.4.2.2 Classe AhcArchivoDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_ARCHIVO. Els mètodes propis dels que consta són els següents:

- **getActivos()**  
Retorna una llista amb tots els objectes AhcArchivo que estan actius.
- **getExternos()**  
Retorna una llista amb tots els objectes AhcArchivo que són externs.
- **getAll()**  
Retorna una llista amb tots els objectes AhcArchivo.

#### 6.4.2.3 Classe AhcArchivo.java

Aquesta és la classe que tracta els registres de la taula AHC\_ARCHIVO com a objectes. Els atributs que té són els següents: **codigo**, **descripcion**, **externo** i **activo**.

Cada atribut té, tal i com ja s'ha dit, un mètode get i un mètode set, per exemple, **getCodigo()** i **setCodigo(String codigo)**, que retornen o estableixen el valor de l'atribut corresponent. A més, implementa els mètodes que tenen totes les classes del mateix tipus i els propis de la classe *AhcArchivoDM.java*.

#### 6.4.2.4 Classe AhcCambioEstadoDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_CAMBIOESTADO. Cal apuntar que aquesta classe no té mètode *update*, ja que els registres d'aquesta taula no es modifiquen, només s'hi insereixen o eliminen. Els mètodes propis dels que consta són els següents:

- **getByDossier(String dossier)**  
Rep un codi de dossier i llança la consulta *AhcCambioEstadoByDossier*, amb el codi de dossier com a paràmetre, que retorna tots els canvis d'estat per al dossier indicat. Retorna una llista amb tots els objectes AhcCambioEstado que compleixen la condició. No busca al DM perquè les probabilitats que tots els canvis d'estat d'un dossier estiguin a memòria són molt baixes.
- **deleteByDossier(String dossier)**  
Elimina tots els canvis d'estat del dossier passat per paràmetre llançant la sentència *deleteAhcCambioEstadoByDossier*.

#### 6.4.2.5 Classe AHCambioEstado.java

Aquesta és la classe que tracta els registres de la taula AHC\_CAMBIOESTADO com a objectes. Els atributs que té són els següents: **codigo**, **dossier**, **fecha**, **hora**, **usuario**, **archivo**, **archivoFinal**, **estado**, **estadoFinal**, **situacion** i **situacionFinal**, amb els corresponents mètodes *get* i *set*. A més, implementa els mètodes que tenen totes les classes del mateix tipus i els propis de la classe *AhcCambioEstadoDM.java*.

#### 6.4.2.6 Classe AhcCategoriaDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_CATEGORIA. Només implementa els mètodes que són comuns a totes les classes d'aquest tipus.

#### 6.4.2.7 Classe AhcCategoria.java

Aquesta és la classe que tracta els registres de la taula AHC\_CATEGORIA com a objectes. Els atributs que té són els següents: **codigo** i **descripcion**, amb els corresponents mètodes *get* i *set*. La resta de mètodes d'aquesta classe són els són comuns per a les classes del mateix tipus que s'han vist fins ara.

#### 6.4.2.8 Classe AhcDossCentroDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_DOSSCENTRO. Té els mateixos mètodes comuns que les classes anteriors. A més, té el següent mètode:

- **getByCentro(String centro)**

Rep el codi d'un centre i busca a memòria els objectes AhcDossCentro que compleixen que el seu atribut *centro* coincideixi amb el passat per paràmetre. Retorna una llista amb tots els objectes AhcDossCentro que compleixen la condició.

#### 6.4.2.9 Classe AhcDossCentro.java

Aquesta és la classe que tracta els registres de la taula AHC\_DOSSCENTRO com a objectes. Els atributs que té són els següents: **centro** i **tipoDoss**, amb els corresponents mètodes *get* i *set*. La resta de mètodes d'aquesta classe són els són comuns per a les classes del mateix tipus i el mètode particulars de la classe *AhcDossCentroDM.java* *getByCentro()*.

#### 6.4.2.10 Classe AhcDossierDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_DOSSIER. Els mètodes propis dels que consta són els següents:

- **getByPaciente(String paciente)**

Donat un codi de pacient, llança la consulta *AhcDossierByPaciente* i retorna una llista amb tots els dossiers que té el pacient indicat.

· **deleteByPaciente(String paciente)**

En primer lloc, es mira si l'usuari té els permisos necessaris per eliminar dossiers. Si es així, es crida la funció *getByPaciente()*. Per cadascun dels objectes *AhcDossier* que retorna aquesta funció, es crida la funció *deleteByDossier()* de la classe *AhcCambioEstado.java*, passant el codi de l'objecte *AhcDossier* com a paràmetre, de la següent manera *AhcCambioEstado.deleteByDossier(ahcDossier.getCodigo())*.

D'aquesta manera, s'eliminen tots els canvis d'estat que hi havia dels dossiers que s'han d'esborrar.

Després, es llança la consulta d'eliminació *deleteAhcDossierByPaciente*, que elimina tots els dossiers del pacient passat per paràmetre.

· **getByPacienteTipo(String paciente, String tipo)**

Donats un pacient i un tipus de dossier, retorna l'objecte *AhcDossier* que compleix aquestes condicions. Cal remarcar que un pacient només pot tenir un dossier d'un determinat tipus. Per aquest motiu es retorna un objecte *AhcDossier* i no una llista. La sentència que es llança per a obtenir l'objecte és *AhcDossierByPacienteTipo*.

· **crearDossier(String tipo, String paciente, String archivo, String estado, String situacion, String volumen, String observaciones, Usuario usuario, boolean manual)**

Rep els paràmetres necessaris per a crear un dossier. Les dates no les rep, ja que s'agafa la data del sistema.

HCIS permet dues maneres de crear dossiers. O bé manualment, per un usuari, o bé automàticament, per l'aplicació. Per exemple, es pot parametritzar que es creï un dossier per a un pacient en quant ingressi a urgències.

Si *manual* és *true*, s'ha de mirar si l'usuari que vol crear el dossier té permís per a fer-ho.

Si l'usuari té permís, o *manual* val *false*, es comprova que el pacient existeix i, si és així, es mira que el dossier no estigui creat per al pacient indicat, cridant la funció *existeByPacienteTipo()* de la classe *AhcDossier.java*.

Si ja existeix i ho estava fent un usuari, se li envia un avís.

En cas que no existeixi, es crea el nou objecte *AhcDossier* i s'insereix a base de dades. També es crea el primer canvi d'estat. Es crea un objecte *AhcCambioEstado*, indicant que l'estat inicial és *CREAMAN* (creat manualment) o *CREAAUTO* (creat automàticament) i l'estat final és disponible.

· **crearDossierPrestado(String tipo, String paciente, String archivo, String estado, String situacion, String volumen, String observaciones, Usuario usuario, String servicio, String agenda, String personaSoli, String motivo, String unenf, String fechaSoli, String horaSoli)**

HCIS permet parametritzar que es creï automàticament en estat PRESTAT. Per exemple, es pot fer que la primera vegada que un pacient vagi a una visita de Consultes Externes, si no el té, se li creï automàticament el dossier. Com el dossier ja el té el metge, inicialment ja està prestat.

El primer que fa es cridar la funció *crearDossier()*, passant-li els paràmetres rebuts i *manual* a *false*.

Com el dossier ja està prestat, s'han de crear la petició, en estat PRESTADA, i préstec corresponents, fent servir les funcions *crearPeticion()* i *prestar* de la classe *AhcPeticion.java*.

També s'ha de crear un objecte `AhcCambioEstado` per indicar que el dossier ha passat d'estat disponible a prestat.

- **`modificarDossier(String dossier, String tipo, String paciente, String archivo, String estado, String situacion, String volumen, String etiqueta, Usuario usuario)`**

Rep els paràmetres necessaris per a modificar un dossier. Les dates no les rep, ja que la data de creació no varia i com a data d'últim moviment s'agafa la del sistema.

En primer lloc mira si l'usuari té permisos per a fer la modificació de dossiers i, si es així, modifica les dades del dossier i crea un nou `AhcCambioEstado`.

- **`eliminarDossier()`**

Es mira si l'usuari té permisos per eliminar dossiers i, si en té, elimina el dossier que ha cridat la funció.

#### 6.4.2.11 Classe `AhcDossier.java`

Aquesta és la classe que tracta els registres de la taula `AHC_DOSSIER` com a objectes. Té els atributs **`codigo`**, **`tipo`**, **`paciente`**, **`fechaCreacion`**, **`archivo`**, **`estado`**, **`situacion`**, **`volumen`**, **`etiqueta`**, **`observaciones`**, **`fechaUltimo`** i **`usuario`**, amb els corresponents mètodes *get* i *set*. Consta amb la implementació de tots els mètodes de la classe `AhcDossierDM.java`. A més, també té les següents funcions:

- **`dossierDisponibleByPaciente(String paciente, String tipo)`**

Crida la funció *getByPacienteTipo()*. Si aquesta funció no retorna resultat, es retorna *false* directament. En altre cas, si el camp estat de l'objecte retornat (*ahcDossier.getEstado()*) és `DISPONIBLE`, es retorna *true*, sinó es retorna *false*.

- **`existeByPacienteTipo(String paciente, String tipo)`**

Crida la funció *getByPacienteTipo()* i, si aquesta funció retorna resultat, es retorna *true*. En cas contrari es retorna *false*.

#### 6.4.2.12 Classe `AhcErrorPrepDM.java`

És la classe encarregada de gestionar el DM de la taula `AHC_ERRORPREP`. A més dels mètodes comuns que les classes anteriors, té el següent mètode:

- **`getActivos()`**

Retorna una llista amb tots els objectes `AhcErrorPrep` que estan actius.

#### 6.4.2.13 Classe `AhcErrorPrep.java`

Aquesta és la classe que tracta els registres de la taula `AHC_ERRORPREP` com a objectes. Té els atributs **`codigo`**, **`descripcion`** i **`activo`**, amb els corresponents mètodes *get* i *set*. La resta de mètodes d'aquesta classe són els són comuns per a les classes que s'han vist fins ara, a més de la implementació del mètode *getActivos()* de la classe `AhcErrorPrepDM.java`.

#### 6.4.2.14 Classe AHCEstadoDossDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_ESTADODOSS. A més dels mètodes comuns que les classes anteriors, té els següents mètodes:

- **getActivos()**

Retorna una llista amb tots els objectes AHCErrorPrep que estan actius.

- **getByCodigoInt(String codigoInt)**

Llança la consulta AHCEstadosDossByCodigoInt i retorna una llista amb tots els registres (transformats en objectes AHCEstadoDoss) que tenen el camp CODIGO\_INT igual que el valor que s'ha passat per paràmetre.

#### 6.4.2.15 Classe AHCEstadoDoss.java

Aquesta és la classe que tracta els registres de la taula AHC\_ESTADODOSS com a objectes. Té els atributs **codigo**, **descripcion**, **codigoInt** i **activo**, amb els corresponents mètodes *get* i *set*. La resta de mètodes d'aquesta classe són els comuns per a les classes que s'han vist fins ara, a més de la implementació dels mètodes *getActivos()* i *getByCodigoInt()* de la classe AHCEstadoDossDM.java.

#### 6.4.2.16 Classe AHC FusionDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_FUSION. Només té els mètodes comuns a totes les taules gestores de DMs.

#### 6.4.2.17 Classe AHC Fusion.java

Aquesta és la classe que tracta els registres de la taula AHC\_FUSION com a objectes. Té els atributs **codigo**, **usuario**, **fecha**, **dossierOrigen** i **dossierDestino**, amb els corresponents mètodes *get* i *set*. A més d'implementar els mètodes de la classe AHC FusionDM.java, també implementa els següents:

- **getByOrigen(String dossierOrigen)**

Retorna totes les fusions en les que el dossier passat per paràmetre és l'origen. Per a fer-ho, crida la funció *getDatos()* de AHC FusionDM.java, passant el codi del dossier com a paràmetre.

- **getByDestino(String dossierDestino)**

Retorna totes les fusions en les que el dossier passat és el destí. Per a fer-ho, crida la funció *getDatos()* de AHC FusionDM.java, passant el codi del dossier com a paràmetre.

#### 6.4.2.18 Classe AHC HPrestamoDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_HPRESTAMO. Té els següents mètodes propis:

- **getByDossier(String dossier)**

Retorna una llista amb tots els préstecs que ha tingut el dossier passat per paràmetre.

- **getFechaUltimaDevolucion(String dossier)**

Retorna quina va ser l'última vegada que es va retornar el dossier indicat a l'arxiu.

#### 6.4.2.19 Classe AhcHPrestamo.java

Aquesta és la classe que tracta els registres de la taula AHC\_ HPRESTAMO com a objectes. Té els atributs **codigo**, **dossier**, **peticion**, **servicio**, **unenf**, **agenda**, **persona**, **motivo**, **ubicacion**, **fechaPrest**, **usuarioPrest**, **observacionesPrest**, **tipoPrest**, **fechaPrevEntr**, **fechaRec**, **usuarioRec** i **observacionesRec**, amb els corresponents mètodes *get* i *set*. Consta amb la implementació de tots els mètodes de la classe AhcHPrestamoDM.java.

#### 6.4.2.20 Classe AhcMotPrestamoDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_MOTPRESTAMO. A més dels mètodes comuns que les classes anteriors, té els següents mètodes:

- **getActivos()**

Retorna una llista amb tots els objectes AhcMotPrestamo que estan actius.

- **getByCodigoInt(String codigoInt)**

Llança la consulta AhcMotPrestamoByCodigoInt i retorna una llista amb tots els registres (transformats en objectes AhcMotPrestamo) que tenen el camp CODIGO\_INT igual que el valor que s'ha passat per paràmetre.

#### 6.4.2.21 Classe AhcMotPrestamo.java

Aquesta és la classe que tracta els registres de la taula AHC\_ MOTPRESTAMO com a objectes. Té els atributs **codigo**, **descripcion**, **codigoInt**, **diasDev**, **categoria** i **activo**, amb els corresponents mètodes *get* i *set*. La resta de mètodes d'aquesta classe són els són comuns per a les classes que s'han vist fins ara, a més de la implementació dels mètodes *getActivos()* i *getByCodigoInt()* de la classe AhcMotPrestamoDM.java.

#### 6.4.2.22 Classe AhcMotUbiDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_MOT\_UBI. Només té els mètodes comuns a totes les taules gestores de DMs.

#### 6.4.2.23 Classe AhcMotUbi.java

Aquesta és la classe que tracta els registres de la taula AHC\_ MOT\_UBI com a objectes. Té els atributs **motivo** i **ubicacion**, amb els corresponents mètodes *get* i *set*. La resta de mètodes són els comuns que tenen les altres classes similars.

#### 6.4.2.24 Classe AHC\_PeticionDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_PETICION. Els mètodes propis que té són els següents:

- **getByPaciente(String paciente)**

Retorna totes les peticions que ha tingut aquest pacient, cridant la funció *getDatos()*, passant el codi del pacient com a paràmetre.

- **getByPacienteTipo(String paciente, String tipo)**

Retorna totes les peticions que ha tingut el dossier del tipus indicat d'aquest pacient, cridant la funció *getDatos()*, passant el codi del pacient i el tipus del dossier com a paràmetres.

- **getByPacienteTipoEstado(String paciente, String tipo, String estado)**

Retorna totes les peticions que ha tingut el dossier del tipus indicat d'aquest pacient i que estiguin en l'estat que es passa per paràmetre, cridant la funció *getDatos()*, passant el codi del pacient, el tipus del dossier i l'estat com a paràmetres.

- **crearPeticion(String dossier, String motivoSoli, String ubicacionSoli, String categoria, String servSoli, String unenfSoli, String agendaSoli, String personaSoli, String fechaSoli, String horaSoli, String fechaDev, String tipoSol, String archivo, String episodio, String cita, String observaciones, Usuario usuario, String estado, String motError, String motAnula)**

Comprova si l'usuari té permís per a crear peticions i, en cas afirmatiu, la crea amb els paràmetres que rep. La data de petició (atribut *fechaPet*) no es passa per paràmetre perquè s'agafa la del sistema.

- **borrarMasivamente(Vector vPeticiones)**

Rep un vector de peticions i elimina totes les que es pugui (les peticions que estan prestades no es poden eliminar, per exemple).

#### 6.4.2.25 Classe AHC\_Peticion.java

Aquesta és la classe que tracta els registres de la taula AHC\_PETICION com a objectes. Té els atributs **codigo**, **dossier**, **motivoSoli**, **ubicacionSoli**, **categoria**, **servSoli**, **unenfSoli**, **agendaSoli**, **personaSoli**, **fechaSoli**, **horaSoli**, **fechaPet**, **fechaDev**, **tipoSol**, **archivo**, **episodio**, **cita**, **observaciones**, **usuario**, **estado**, **motError** i **motAnula**, amb els corresponents mètodes *get* i *set*. A banda de la implementació de tots els mètodes de la classe AHC\_PeticionDM.java, també té els següents mètodes:

- **eliminar(Usuario usuario)**

Mira si l'usuari té permisos per eliminar peticions i si la petició es pot eliminar (només es poden eliminar peticions que estiguin sense preparar o que estiguin preparades amb error) i, si es així, posa la variable *\_eliminado* a *true* i crida el mètode *salvar()* per a l'objecte que està cridant aquesta funció.

- **preparar(Usuario usuario)**

L'acció de preparar és mirar on està el dossier, per saber si està en condicions de ser prestat.



Primer de tot, es mira si l'usuari té permisos per a preparar peticions. Si en té, es comprova que el dossier per al qual s'està fent la petició existeixi. En cas de que no existeixi, la petició es quedarà en estat PREPARADA AMB ERROR i el motiu de l'error serà que el dossier no existeix.

Si el dossier existeix, es mirarà que estigui a l'arxiu i, en cas de que així sigui, la petició quedarà PREPARADA CORRECTAMENT. En cas contrari, la petició també quedarà PREPARADA AMB ERROR i el motiu d'error serà el que correspongui, ja sigui perquè el dossier està prestat, o perquè s'ha extraviat, o perquè està al CLS, per exemple.

· **prestar(Usuario usuario)**

En primer lloc es comprova que l'usuari tingui permisos per a prestar dossiers i que la petició es pugui prestar.

En aquest cas, es canvia l'estat de la petició a PRESTADA, crea el nou préstec (tant l'objecte AhcPrestamo com l'AhcHPrestamo) i modifica el dossier, que passa a tenir estat PRESTAT.

Finalment, es tornen a preparar la resta de peticions que tenia el mateix dossier i que estaven en estat PREPARADES CORRECTAMENT. Com el dossier ja es troba prestat, aquestes altres peticions quedaran en estat PREPARADA AMB ERROR i el motiu d'error serà DOSSIER PRESTAT.

· **recuperar(Usuario usuario)**

Després de comprovar que l'usuari té permisos per a recuperar dossiers, es canvia l'estat de la petició a RECUPERADA, el dossier torna a estar DISPONIBLE, s'elimina el préstec de la taula AHC\_PRESTAMO i s'actualitza el camp que indica la data de devolució al registre corresponent de la taula AHC\_HPRESTAMO.

· **anular(Usuario usuario, String motivo)**

Si l'usuari té permisos per anul·lar peticions, i la petició que crida la funció es pot anul·lar, se li canvia l'estat a ANUL·LADA i se li especifica el motiu passat per paràmetre.

· **modificar(String dossier, String motivoSoli, String ubicacionSoli, String categoria, String servSoli, String unenfSoli, String agendaSoli, String personaSoli, String fechaSoli, String horaSoli, String fechaDev, String tipoSol, String archivo, String episodio, String cita, String observaciones, Usuario usuario, String estado, String motError, String motAnula)**

Si l'usuari té permisos per a modificar peticions, es crida als corresponents mètodes *set* de cada atribut i es crida la funció *salvar()*.

· **calcularFechaDevolucion(AhcMotPrestamo ahcMotPrestamo, String fechaSoli, Usuario usuario)**

La data prevista de devolució va en funció del motiu de préstec. Aquesta funció s'encarrega de calcular-la.

#### 6.4.2.26 Classe AhcPrestamoDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_PRESTAMO. Té els següents mètodes propis:

- **getByDossier(String dossier)**

Retorna el préstec actiu que té el dossier que es passa per paràmetre, cridant la funció *getDatos()*, passant el codi del dossier com a paràmetre.

- **getByPeticones(List IPeticiones)**

Retorna els préstecs actius associats a les peticions que es passen a la llista rebuda per paràmetre, cridant la funció *getDatos()*, passant els codis de les peticions separats per comes (,) com a paràmetre.

#### 6.4.2.27 Classe AhcPrestamo.java

Aquesta és la classe que tracta els registres de la taula AHC\_PRESTAMO com a objectes. Té els atributs **codigo**, **dossier**, **peticion**, **servicio**, **unenf**, **agenda**, **persona**, **motivo**, **ubicacion**, **fecha**, **usuario**, **observaciones**, **tipo** i **fechaPrevDev**, amb els corresponents mètodes *get* i *set*. Consta amb la implementació de tots els mètodes de la classe AhcPrestamoDM.java. A més, també implementa els següents:

- **esMoroso()**

Retorna *true* si l'atribut *fechaPrevDev* no és nul i aquesta data és anterior a la data actual.

- **rectificar(String dossier, String peticion, String servicio, String unenf, String agenda, String persona, String motivo, String ubicacion, String observaciones, String tipo, String fechaPrevDev)**

Actualitza les dades del préstec que crida la funció (i de la petició que té associada) amb les dades que es passen per paràmetre.

- **modificar(String dossier, String peticion, String servicio, String unenf, String agenda, String persona, String motivo, String ubicacion, String observaciones, String tipo, String fechaPrevDev)**

Recupera el préstec i la petició que té associada, i en crea un de nou, amb una petició relacionada, amb les dades que es passen per paràmetre.

- **eliminar(Usuario usuario)**

Mira si l'usuari té permisos per eliminar préstecs i, si es així, posa la variable *\_eliminado* a *true* i crida el mètode *salvar()* per a l'objecte que està cridant aquesta funció.

#### 6.4.2.28 Classe AhcSituacionDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_SITUACION. Només té els mètodes comuns a totes les taules gestores de DMs.

#### 6.4.2.29 Classe AhcSituacion.java

Aquesta és la classe que tracta els registres de la taula AHC\_SITUACION com a objectes. Té els atributs **codigo** i **descripcion**, amb els corresponents mètodes *get* i *set*. La resta de mètodes són els comuns que tenen les altres classes similars.

#### 6.4.2.30 Classe AhcTipoDossDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_TIPODOSS. A més dels mètodes comuns que les classes anteriors, té el següent mètode:

- **getActivos()**

Retorna una llista amb tots els objectes AhcTipoDoss que estan actius.

#### 6.4.2.31 Classe AhcTipoDoss.java

Aquesta és la classe que tracta els registres de la taula AHC\_TIPODOSS com a objectes. Té els atributs **codigo**, **descripcion** i **activo**, amb els corresponents mètodes *get* i *set*. La resta de mètodes d'aquesta classe són els són comuns per a les classes que s'han vist fins ara, a més de la implementació del mètode *getActivos()* de la classe AhcTipoDossDM.java.

#### 6.4.2.32 Classe AhcUbicacionDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_UBICACION. Només té els mètodes comuns a totes les taules gestores de DMs.

#### 6.4.2.33 Classe AhcUbicacion.java

Aquesta és la classe que tracta els registres de la taula AHC\_UBICACION com a objectes. Té els atributs **codigo** i **descripcion**, amb els corresponents mètodes *get* i *set*. La resta de mètodes són els comuns que tenen les altres classes similars.

#### 6.4.2.34 Classe AhcVolumenDM.java

És la classe encarregada de gestionar el DM de la taula AHC\_VOLUMEN. A més dels mètodes comuns que les classes anteriors, té els següents mètodes:

- **getByDossierVolumen(String dossier, String volumen)**

Retorna l'objecte AhcVolumen que pertany al dossier passat per paràmetre i que té el número de volum que s'indica. Per fer-ho, es crida la funció *getDatos()*, passant el codi del dossier i el número de volum per paràmetre.

- **getByDossier(String dossier)**

Retorna una llista amb tots els volums que té el dossier que es passa per paràmetre, cridant la funció *getDatos()*, passant el codi del dossier com a paràmetre.

#### 6.4.2.35 Classe AhcVolumen.java

Aquesta és la classe que tracta els registres de la taula AHC\_VOLUMEN com a objectes. Té els atributs **codigo**, **dossier**, **volumen**, **fechaCreac**, **estado** i **observaciones**, amb els corresponents mètodes *get* i *set*.

A banda de la implementació de tots els mètodes de la classe AhcVolumenDM.java, també té el següents mètode:

· **eliminar(Usuario usuario)**

Mira si l'usuari té permisos per eliminar volums i si el volum es pot eliminar (els volums que estan prestats no es poden eliminar fins que no s'han recuperat) i, si es així, posa la variable *\_eliminado* a *true* i crida el mètode *salvar()* per a l'objecte que està cridant aquesta funció.

## 6.5 JSPs i classes de les pantalles

En aquest apartat es descriurà quin sistema fa servir HCIS per a mostrar la informació en les diferents pantalles de les que consta l'aplicació.

Cal dir que totes les pantalles segueixen el mateix patró, així que no s'explicaran totes detingudament, sinó que es farà una visió general de quines tècniques se segueixen per a interactuar amb l'usuari.

### 6.5.1 El fitxer *actions.properties*

Per fer crides a classes Java des de codi HTML (incrustat en pàgines JSP) es fan servir *accions*. Una acció no és més que un àlies d'una classe Java, que es fa servir perquè les classes no es poden cridar directament des de codi HTML. Així, des del codi HTML s'invoca l'acció que pertorqui, i aquesta s'encarregarà d'executar la classe corresponent.

El mapeig de les accions amb les classes Java es troba al fitxer de propietats *actions.properties*.

D'aquesta manera si, per exemple, des de la pantalla de peticions es vol que es mostrin unes dades en funció d'uns criteris de cerca introduïts, al pulsar el botó de *Cerca*, s'invocarà a l'acció *buscarAhcPeticiones.do*. Al cridar a aquesta acció, el que s'està fent en realitat és cridar a la classe *BuscarAhcPeticiones.java*.

Amb aquest sistema s'aconsegueix cridar a classes Java des de codi HTML.

### 6.5.2 Les pantalles

Quan es tria una opció del menú que hi ha a la part esquerra de l'aplicació, el que es fa és carregar-se una pantalla a la part dreta, tal i com es pot veure a la Figura 6.1., on s'ha triat l'opció *Gestió de Préstecs* del submenú d'arxiu.

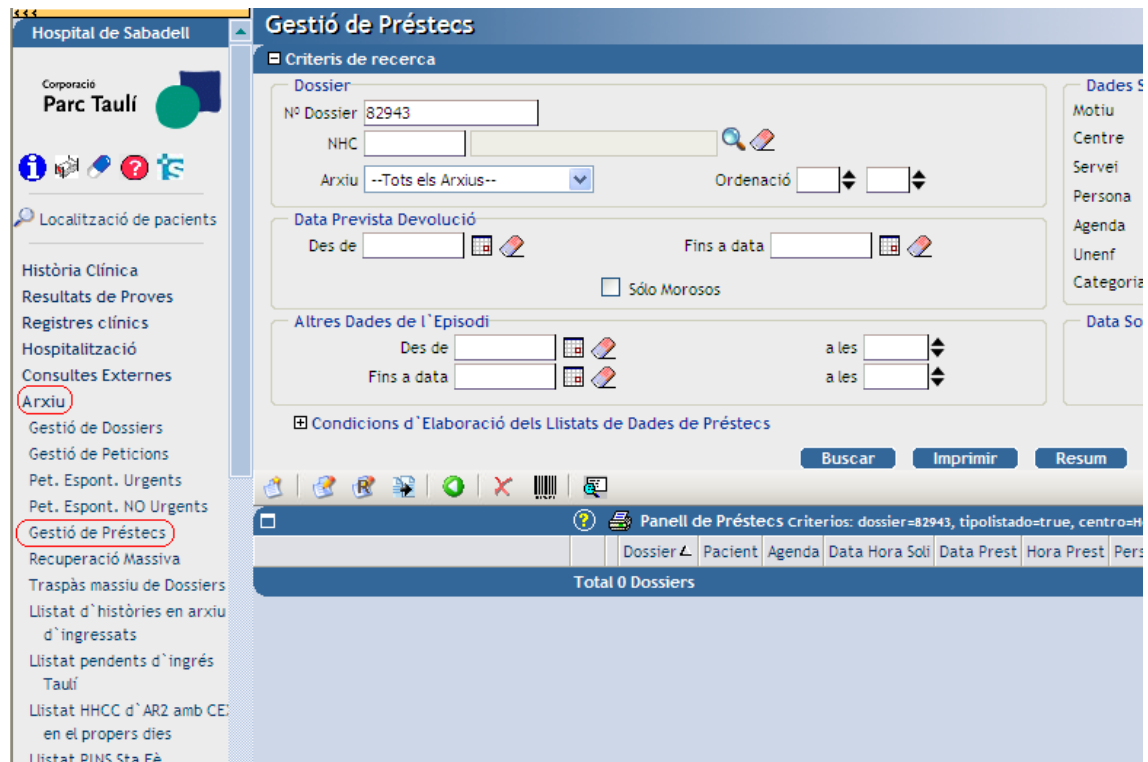


Figura 6.1. Obrir una pantalla des del menú

Aquestes pantalles consten d'un formulari a la meitat superior i d'una part destinada a mostrar els resultats de les cerques, que inicialment està buida. A més, separant ambdues parts de la pantalla, hi ha una fila de botons (botonera) que serveixen per a realitzar accions sobre l'element seleccionat de la part inferior, un cop s'ha realitzat una cerca.

Com s'acaba de dir, quan es llança una cerca, els resultats de la mateixa es mostren en la part inferior de la pantalla. En canvi, les accions que es realitzen amb els botons que hi ha la botonera sobre un dels elements obtinguts de la cerca, obren una nova finestra.

Hi ha, però, una excepció. Les pantalles que es fan servir per al manteniment de les taules mestres ja executen una acció en carregar-se, mostrant sempre tot el contingut de la taula.

### 6.5.3 Els panells

Les meitats inferiors de les pantalles, és a dir, on es mostren els resultats de les cerques, dintre d'HCIS s'anomenen *panells*.

Els panells els formen una classe Java i una pàgina JSP. Quan es polsa el botó de *Buscar* a una pantalla, s'invoca l'acció corresponent. Aquesta acció obtén les dades que calguin i crea una instància d'un objecte de la classe corresponent al panell que pertoqui. Un cop creat aquest objecte panell, des de l'acció es crida al JSP corresponent, que s'encarrega de mostrar les dades que conté l'objecte panell que s'acaba de crear.

Sobre cadascun dels resultats mostrats en el panell és sobre qui es podran executar les accions de la botonera.

Seguint amb el mateix exemple d'abans, quan, des de la pantalla de gestió de peticions, *pantallaAhcPeticiones.jsp*, s'invoca l'acció *buscarAhcPeticiones.do*, el que fa la classe *BuscarAhcPeticiones.java* és agafar tots els paràmetres que s'ha introduït al formulari de la pantalla, crear un objecte *PanelAhcPeticiones* i cridar la funció *inicializar()* de la classe *PanelAhcPeticiones.java*, passant-li els paràmetres rebuts. Aquesta funció obté totes les peticions que compleixin les característiques indicades i les emmagatzema a l'atribut *datos* de l'objecte *PanelAhcPeticiones*. La classe *BuscarAhcPeticiones.java* crida el JSP *panelAhcPeticiones.jsp* passant-li el nou objecte com a paràmetre. Aquest JSP s'encarrega de mostrar els resultats de la cerca.

#### 6.5.4 Les classes Ajax

Per acabar amb els panells, comentar que també es fa servir la tecnologia *Ajax* (acrònim de *Asynchronous JavaScript And XML*)[15]. Aquesta tecnologia es fa servir per a calcular dades en funció d'altres dades en codi HTML sense haver de recarregar la pantalla. Per exemple, si, dintre d'un formulari, hi ha dues dades que s'han de seleccionar amb una llista desplegable, però els valors de la segona llista depenen del que s'hagi seleccionat a la primera, una possible solució seria, un cop se selecciona el primer valor, tornar a refrescar la pàgina, tenint en compte el valor que s'acaba de marcar i calculant els valors de la segona llista en funció d'aquest.

Però això donaria un aspecte no gaire agradable de cara a l'usuari, ja que, si hi hagués més d'un desplegable d'aquest tipus, s'estaria refrescant constantment la pantalla, impedit-li treballar amb comoditat. Així, des de codi HTML es pot cridar a classes Ajax que permeten realitzar aquestes accions sense que l'usuari se n'adoni. És a dir, calcular els valors que s'han de mostrar a la segona llista en funció del valor que s'hagi seleccionat a la primera, però sense la necessitat de refrescar la pàgina. D'aquesta manera s'incrementa la interactivitat, velocitat i la facilitat d'ús de l'aplicació.

Un exemple pot ser a l'hora de modificar les dades d'un dossier, ja que, en funció de l'arxiu que se seleccioni, s'han de mostrar unes opcions o unes altres als desplegables de situació i estat del dossier, ja que, si l'arxiu és CLS, per exemple, només pot estar en situació *SEMIPASSIU* o *EXITUS* i no pot estar en estat *DISPONIBLE* o *PRESTADO*.

### 6.6 Les accions

Tal i com s'ha comentat a l'apartat anterior, per poder cridar classes Java des de pàgines web es fan servir accions. En aquest apartat es descriurà què realitza cada acció i des d'on es fa servir. Es diferenciaren les accions que executa el mòdul AHC en dos tipus, les que actuen sobre un dossier (crear, modificar, prestar, etc), i les que es fan servir des dels manteniments de les taules mestres (crear, modificar, eliminar objectes *AhcArchivo*, per exemple).

## 6.6.1 Les accions que actuen sobre dossiers

En aquest apartat es descriurà el funcionament de les accions del primer tipus. No es profunditzarà al detall, com s'ha fet amb les classes que s'usen per a interactuar amb la base de dades, sinó que se'n farà una visió general de què fan i per a què es fan servir. Totes aquestes classes deriven de la classe *Action.java* i implementen el mètode *perform()*, que és el que s'executa en invocar l'acció corresponent.

El mòdul AHC té quatre pantalles principals: **Gestió de Dossiers**, **Gestió de Peticions**, **Gestió de Préstecs** i **Traspàs de dossiers**. Per tal d'agrupar les accions, s'explicaran les classes segons la pantalla des de la que es fan servir.

### 6.6.1.1 La classe *BuscarAhcDossier.java*

Quan s'accedeix a la pantalla de *Gestió de Dossiers*, inicialment està buida i hi ha un camp de text per a introduir un número d'història clínica d'un pacient. En introduir un número i pulsar *INTRO* s'executa aquesta acció que s'encarrega de buscar tots els dossiers que té el pacient i mostrar les dades del que té el menor codi de tipus de dossier, és a dir, si el pacient té dossiers de tipus 0 i 1, per exemple, es carreguen les dades del dossier de tipus 0.

### 6.6.1.2 La classe *NuevoAhcDossier.java*

És l'acció que es crida quan es clica el botó de crear un nou dossier un cop ja s'ha introduït un número d'història clínica.

En primer lloc, comprova que l'usuari tingui privilegis per a crear dossiers. Si és així, calcula, pel pacient indicat, quins tipus de dossier ja té creats, per permetre crear només dossiers dels tipus dels que encara no en té el pacient.

Si el pacient ja té tots els tipus de dossier, es mostra un avís a l'usuari. En cas contrari, obre la finestra de creació de dossier, permetent crear només dossiers dels tipus que li manquin al pacient.

### 6.6.1.3 La classe *CrearAhcDossier.java*

Un cop ja s'han seleccionat totes les dades necessàries per a crear un nou dossier es prem el botó *Acceptar*, que crida a aquesta acció, que és l'encarregada de crear el nou objecte *AhcDossier*.

### 6.6.1.4 La classe *ModificarAhcDossier.java*

Tenint un dossier seleccionat, si es pitja el botó de modificar, es crida aquesta acció.

El primer que es fa és comprovar que l'usuari tingui privilegis per a modificar dossiers. En cas de que així sigui, s'obtenen les dades del dossier que es vol modificar per a poder-les carregar a la pantalla de modificació de dossiers.

### **6.6.1.5 La classe ActualizarAhcDossier.java**

Quan ja s'han introduït totes les noves dades del dossier i es polsa el botó *Aceptar*, es crida aquesta acció per a guardar-les. Si l'estat del dossier és *PRESTADO*, no permet canviar-lo des d'aquesta pantalla. S'ha de retornar primer a l'arxiu, mitjançant la recuperació del préstec actiu que tingui el dossier.

### **6.6.1.6 La classe EliminarAhcDossier.java**

Amb un dossier seleccionat, si es polsa el botó d'eliminar dossier, s'executa aquesta acció.

Si l'usuari té privilegis per a eliminar dossiers, es comprova que el dossier es pugui eliminar, ja que un dossier que hagi tingut peticions, encara que ja estiguin retornades, no es pot eliminar. Així, l'acció d'eliminar dossier només s'utilitza per a eliminar dossiers que s'acaben de crear per error i que, per tant, encara no han tingut moviment. Si el dossier es pot eliminar, es fa.

### **6.6.1.7 La classe ConsultarAhcVolumenes.java**

Tenint un dossier seleccionat, si es clica sobre el botó de veure els volums d'aquest dossier, es crida aquesta acció que busca tots els volums del dossier marcat i els mostra per pantalla.

### **6.6.1.8 La classe MostrarAhcVolumenes.java**

Mentre s'estan modificant les dades d'un dossier, una de les accions que es pot fer es veure quins volums té, però podent crear-ne de nous, modificar les dades dels que ja existeixen o eliminar-los.

### **6.6.1.9 La classe NuevoAhcVolumen.java**

Quan es clica el botó corresponent, si l'usuari té els privilegis necessaris, es crida aquesta classe que obre la finestra de crear nou volum, passant-li el codi del dossier al que pertanyerà.

### **6.6.1.10 La classe CrearAhcVolumen.java**

Quan ja s'han introduït les dades corresponents al volum i es prem el botó *Aceptar*, es crida aquesta acció, que és l'encarregada de crear el nou objecte *AhcVolumen*.

### **6.6.1.11 La classe ModificarAhcVolumen.java**

Clicant el botó adequat, es crida aquesta acció, que obre una finestra amb les dades del volum que es vol modificar.



#### **6.6.1.12 La classe ActualizarAhcVolumen.java**

Quan ja s'han introduït totes les noves dades del volum i es polsa el botó *Acceptar*, es crida aquesta acció per a guardar-les. L'acció s'executa sempre i quan l'usuari tingui privilegis per fer-ho.

#### **6.6.1.13 La classe EliminarAhcVolumen.java**

Amb un volum seleccionat, si es polsa el botó d'eliminar volum, s'executa aquesta acció, si l'usuari té els privilegis necessaris.

#### **6.6.1.14 La classe MostrarImpresionEtiquetas.java**

Una altra de les accions que es pot fer és imprimir les etiquetes que després s'enganxen als dossiers per tal d'identificar-los. Quan es clica el botó d'impressió d'etiquetes es crida aquesta acció, que obre la finestra on s'introduiran els codis dels dossiers dels que es vol imprimir l'etiqueta.

#### **6.6.1.15 La classe ImprimirEtiquetas.java**

A l'hora d'imprimir etiquetes, els codis es poden introduir de tres formes: o bé es pot introduir un sol número de dossier, o bé es poden introduir diversos números separats per comes (,), o bé es pot introduir un rang separat per dos punts (:).

Si l'usuari té privilegis per a fer-ho, es calculen totes les etiquetes que s'han d'imprimir, en cas que s'hagi d'imprimir més d'una, i s'imprimeixen d'una en una. Si, quan es va a imprimir una etiqueta, el dossier associat encara no existeix, també el crea.

#### **6.6.1.16 La classe BuscarAhcPetition.java**

Quan s'accedeix a la pantalla de *Gestió de Peticions*, inicialment es mostra el panell buit. A la part superior de la pantalla hi ha una sèrie de filtres per a acotar les peticions que es vol que es mostrin al panell inferior. Un cop s'han introduït els filtres desitjats i s'ha pitjat el botó *Buscar*, es llança aquesta acció, que s'encarrega de mostrar per pantalla totes les peticions que compleixin els criteris desitjats.

#### **6.6.1.17 La classe PrepararAhcPetition.java**

És l'acció que es llança si, un cop introduïts els filtres desitjats, es clica el botó *Preparar* en comptes del botó *Buscar*. El que fa aquesta acció es buscar totes les peticions que compleixen els criteris introduïts però, en comptes de mostrar-les per pantalla, les prepara. És a dir, mira on es troba el dossier de cada petició.

L'acció de preparar només s'executa sobre les peticions que estan en estat *INICIAL* o *PREPERERROR* (preparat amb error). Si el dossier de la petició es troba a l'arxiu, la petició queda en estat *PREPOK* (preparada correctament). Sinó, queda en estat *PREPERERROR*, marcant el motiu de préstec que correspongui.

Quan es llança la preparació de peticions, no es mostren els resultats per pantalla, ja que normalment es preparen moltes peticions de cop i no resulta pràctic que es mostrin moltes peticions per pantalla (milers en molts casos).

#### **6.6.1.18 La classe NuevaAhcPetition.java**

Com ja s'ha explicat anteriorment, les peticions es poden crear de dues formes diferents. Automàticament, quan es programen visites per a determinades agendes de Consultes Externes, per exemple, o manualment. Les peticions manuals són aquelles que les fa explícitament l'usuari.

Quan es polsa el botó de crear nova petició, es crida aquesta acció, que s'encarrega de mirar que l'usuari tingui privilegis per a crear peticions de dossier, i sobre quins tipus de dossier té permisos per a crear peticions.

En cas que l'usuari pugui crear peticions, s'obre la finestra de creació de peticions manuals. Al desplegable de tipus de dossier només hi apareixeran aquells tipus sobre els que l'usuari té permís per a crear peticions.

Aquesta acció, a més de des de la pantalla de Gestió de Peticions, també s'ha habilitat des d'altres llocs de l'aplicació, ja que no només el personal d'arxiu pot fer peticions. Per exemple, un metge ha de poder demanar dossiers per a fer estudis, però no ha de tenir accés al mòdul AHC.

#### **6.6.1.19 La classe CrearAhcPetition.java**

Quan ja s'han introduït les dades necessàries per a realitzar la nova petició, es pitja el botó *Acceptar* o *Acceptar i Nova*. Ambdós botons criden aquesta acció, que és la que s'encarrega de crear la nova petició. La diferència és que, si s'ha clicat el botó *Acceptar*, aquesta acció tancarà la finestra de nova petició un cop s'hagi creat, mentre que, si s'ha premut el botó *Acceptar i Nova*, un cop s'hagi creat la petició, en comptes de tancar-se la finestra, aquesta quedarà oberta amb les dades carregades excepte el número de pacient sobre el que es realitza la petició. Això es força útil ja que, seguint amb l'anterior exemple, si un metge ha de demanar dossiers per a fer un estudi, tots els dossiers tindran el mateix motiu de sol·licitud, s'hauran de portar a la mateixa ubicació, etc. D'aquesta manera no s'han d'introduir totes les dades per a cada dossier.

#### **6.6.1.20 La classe ModificarAhcPetition.java**

Després de buscar peticions, tenint-ne una seleccionada, si es pitja el botó de modificar, es crida aquesta acció.

El primer que es fa és comprovar que l'usuari tingui privilegis per a modificar peticions. En cas de que així sigui, s'obtenen les dades de la petició que es vol modificar per a poder-les carregar a la pantalla de modificació de peticions. Només es poden modificar, però, les peticions que estan en estat *INICIAL* o *PREPERROR*.

#### **6.6.1.21 La classe ActualizarAhcPeticion.java**

Quan ja s'han introduït totes les noves dades de la petició i es polsa el botó *Acceptar*, es crida aquesta acció per a guardar-les.

#### **6.6.1.22 La classe EliminarAhcPeticion.java**

Amb una (o vàries) petició seleccionada, si es polsa el botó d'eliminar petició, s'executa aquesta acció.

Si l'usuari té privilegis per a eliminar peticions, elimina les peticions seleccionades que es trobin en estat *INICIAL* o *PREPERROR*.

#### **6.6.1.23 La classe PrepararAhcPeticionesSeleccionadas.java**

Igual que, quan es polsa el botó de *Preparar*, es preparen totes les peticions que compleixin els criteris introduïts, després de fer una cerca també es poden preparar només unes peticions concretes.

Selecciónant les peticions desitjades i clicant el botó corresponent, s'executa aquesta acció que, si l'usuari té privilegis, prepara totes les peticions seleccionades que estiguin en estat *INICIAL* o *PREPERROR*.

#### **6.6.1.24 La classe PrestarAhcPeticion.java**

Quan es mostra la finestra per a prestar o recuperar peticions, si es tria l'opció *Prestar*, per a cada dossier que es llegeixi amb la pistola, s'executarà aquesta acció.

Per a poder prestar peticions, s'ha de realitzar prèviament una cerca de peticions, i només es podran prestar peticions que apareguin en els resultats de la cerca i que estiguin en estat *PREPOK*, per tal de saber quina de les peticions que té un dossier és la que s'ha de prestar, si és que en té més d'una. En cas que hi hagi més d'una petició per al mateix dossier en els resultats de la cerca, al llegir l'etiqueta amb la pistola lectora de codis de barres, la petició que quedarà prestada serà la que s'hagi de servir primer, és a dir, la que tingui una data més propera a l'actual. La cerca es realitza per evitar, en el major nombre de casos, el fet d'haver de buscar quina és la primera petició que s'ha de servir per al dossier introduït, ja que aquest procés fa que l'acció de prestar sigui més lenta. Aquesta acció s'executa sempre i quan l'usuari tingui privilegis per a prestar peticions.

#### **6.6.1.25 La classe RecuperarAhcPeticion.java**

Si s'escull l'opció *Recuperar*, es recuperarà la petició prestada del dossier que s'hagi llegit. Com només hi pot haver un préstec actiu per a un dossier, no hi ha conflicte sobre quin s'ha de recuperar. Per aquest motiu, si l'acció que es vol fer és la de recuperar, no cal realitzar una cerca prèvia.

Com la resta d'accions, l'usuari ha de tenir uns determinats privilegis per a poder executar aquesta.

#### **6.6.1.26 La classe PrestarAhcPeticionesSeleccionadas.java**

Si el que es vol fer és prestar peticions determinades que hi ha als resultats d'una cerca, es pot fer seleccionant-les i clicant el botó de prestar peticions seleccionades.

Si l'usuari té privilegis per a prestar peticions, s'executa aquesta acció que presta totes les peticions seleccionades que estiguin en estat *PREPOK*.

#### **6.6.1.27 La classe PedirAhcDossierACLS.java**

Com s'ha comentat en capítols anteriors, la CSPT consta de tres Arxius d'Històries Clíniques. Dos d'interns que es troben a la mateixa Corporació i un d'extern que es troba a la localitat de Santa Perpètua de Mogoda, i que anomenen CLS.

El personal de CLS no fa servir HCIS, per tant, se'ls ha d'informar d'alguna manera de quines històries es necessitaran, per tal que les puguin enviar i així estiguin disponibles per a ser prestades.

Per a identificar quines peticions necessiten un dossier que està al CLS, en el moment de crear-se la petició, si el dossier es troba a l'arxiu extern, el camp *ARCHIVO* de la petició serà CLS. A més, aquests dossiers estan en estat *ALP*, que és un estat que indica que aquests dossiers es troben al CLS, però que tenen peticions pendents.

D'aquesta manera, quan s'han de demanar dossiers al CLS, s'han de seleccionar les peticions desitjades i clicar el botó corresponent, que executa aquesta acció.

Si l'usuari té privilegis, es crea un fitxer que inclou els codis dels dossiers que corresponen a les peticions seleccionades i que es troben en estat *ALP*. Aquest fitxer té un format específic i s'envia a un directori determinat del servidor *cls.cspt.es*, on el personal de CLS l'anirà a buscar per a processar-lo i saber quins dossiers ha d'enviar cap a la CSPT.

A més, els dossiers que s'han inclòs al fitxer passen a estar en estat *ALT*, que indica que la petició al CLS ja s'ha tramitat.

#### **6.6.1.28 La classe RecibirAhcDossierDeCLS.java**

Quan el personal de CLS processa el fitxer enviat, troba els dossiers que s'han sol·licitat i els envia, genera un altre fitxer amb tots els dossiers que van des de CLS cap a la CSPT, que també deixa al servidor *cls.cspt.es*.

Aquesta acció s'encarrega de llegir el fitxer enviat des del CLS, sempre i quan l'usuari tingui privilegis per a executar-la, i posar els dossiers que conté en estat *ALC*, que és un estat que indica que l'enviament del dossier ja s'ha confirmat des del CLS.

#### **6.6.1.29 La classe RecepcionarAhcDossierDeCLS.java**

Quan arriben els dossiers de CLS, hi ha una nova finestra per a llegir etiquetes de codis de barres. Cada dossier llegit es passa a estat *DISPONIBLE*.

### 6.6.1.30 La classe **BuscarAhcPrestamo.java**

Al accedir a la pantalla de *Gestió de Préstecs*, inicialment es mostra el panell buit. A la part superior de la pantalla hi ha una sèrie de filtres per a acotar els préstecs que es vol que es mostrin al panell inferior. Un cop s'han introduït els filtres desitjats i s'ha pitjat el botó *Buscar*, es llança aquesta acció, que s'encarrega de mostrar per pantalla totes els préstecs que compleixin els criteris desitjats.

### 6.6.1.31 La classe **NuevoAhcPrestamo.java**

Hi ha vegades que s'han de prestar dossiers sense que se li hagi realitzat una petició, com pot ser un pacient que entra a Urgències i es necessita el seu dossier immediatament.

Quan es polsa el botó de crear nou préstec, es crida aquesta acció, que s'encarrega de mirar que l'usuari tingui privilegis per a crear préstecs de dossier, i sobre quins tipus de dossier té permisos per a crear préstecs.

Si l'usuari pot crear préstecs, s'obre la finestra de creació de préstecs manuals. Al desplegable de tipus de dossier només hi apareixeran aquells tipus sobre els que l'usuari té permís per a crear préstecs.

Quan se selecciona el motiu de préstec, es comprova si el dossier té alguna petició pendent per a aquest motiu i, si es així, s'obre una nova finestra per a seleccionar la petició que es desitgi, si és que és alguna de les que es mostra. Si se selecciona alguna petició, la resta de valors es carreguen en funció del que tingui aquesta. Si no se selecciona cap petició, la resta de valors queden en blanc.

### 6.6.1.32 La classe **CrearAhcPrestamo.java**

Quan ja s'han introduït les dades necessàries per a realitzar el nou préstec, es pitja el botó *Acceptar* o *Acceptar i Nou*.

Polsant aquests botons es crida aquesta acció, que, si l'usuari té privilegis, s'encarrega de crear una nova petició pel dossier seleccionat amb les dades introduïdes, prepara la petició i la presta. Si el dossier seleccionat ja es troba prestat, no se li pot crear un nou préstec, havent-se de recuperar el préstec actiu.

La diferència que hi ha entre els dos botons és que, si s'ha clicat el botó *Acceptar*, aquesta acció tancarà la finestra de nou préstec un cop s'hagi creat, mentre que, si s'ha premut el botó *Acceptar i Nou*, un cop s'hagi creat el préstec, en comptes de tancar-se la finestra, aquesta quedarà oberta amb les dades carregades excepte el número de pacient sobre el que es realitza el préstec.

### 6.6.1.33 La classe **RectificarAhcPrestamo.java**

Tenint un préstec dels que ha retornat la cerca, si es pitja el botó de rectificar, es crida aquesta acció.

El primer que es fa és comprovar que l'usuari tingui privilegis per a rectificar préstecs. En cas de que així sigui, s'obtenen les dades del préstec que es vol rectificar per a poder-les carregar a la pantalla de rectificació de préstecs.

#### **6.6.1.34 La classe ActualizarAhcPrestamo.java**

Quan ja s'han introduït totes les noves dades del préstec i es polsa el botó *Aceptar*, es crida aquesta acció per a guardar-les. També es modifiquen les dades de la petició associada al préstec.

#### **6.6.1.35 La classe ModificarAhcPrestamo.java**

Tenint un préstec dels que ha retornat la cerca, si es pitja el botó de modificar, es crida aquesta acció.

El primer que es fa és comprovar que l'usuari tingui privilegis per a modificar préstecs. En cas de que així sigui, s'obtenen les dades del préstec que es vol modificar per a poder-les carregar a la pantalla de modificació de préstecs.

#### **6.6.1.36 La classe ModificacionAhcPrestamo.java**

Quan ja s'han introduït totes les noves dades del préstec i es polsa el botó *Aceptar*, es crida aquesta acció, que s'encarrega de recuperar el préstec seleccionat i crear-ne un de nou, amb la corresponent petició associada.

#### **6.6.1.37 La classe EliminarAhcPrestamo.java**

Si l'usuari té privilegis per a eliminar préstecs, quan es polsa el botó d'eliminar préstecs, s'eliminen tots els préstecs seleccionats, així com les peticions associades a cadascun. Els dossiers queden en estat *DISPONIBLE*.

#### **6.6.1.38 La classe RecuperarAhcPrestamo.java**

Tenint un préstec seleccionat, si es clica el botó de recuperar préstec, en cas de que l'usuari tingui privilegis per a recuperar préstecs, es crida aquesta acció que s'encarrega d'obrir una nova finestra amb les dades del préstec, permetent introduir noves observacions de recuperació, ja que els préstecs que es recuperen des d'aquí acostumen a ser casos particulars. El cas normal és que es recuperin des de la *Gestió de Peticions*, amb la pistola lectora de codis de barres.

#### **6.6.1.39 La classe RecuperacionAhcPrestamo.java**

Si es clica el botó d'*Aceptar* a la finestra de recuperació de préstecs, s'executa aquesta acció que, com el seu nom indica, s'encarrega de recuperar el préstec indicat, marcant la petició associada com a *RECUPERADA* i passant el dossier a estat *DISPONIBLE*.

#### 6.6.1.40 La classe **BuscarAhcDossieres.java**

A part de la *Gestió de Dossiers*, que només mostra informació dels dossiers del pacient seleccionat, hi ha la pantalla de *Traspàs de Dossiers*. Un traspàs no és més que un canvi d'arxiu.

Aquesta pantalla és més similar a la de *Gestió de Peticions* o *Gestió de Préstecs*, ja que presenta una sèrie de filtres que permeten buscar els dossiers que compleixin una sèrie de condicions.

Aquesta és l'acció que s'executa al pitjar el botó *Buscar* a l'anomenada pantalla de *Traspàs de Dossiers*, si l'usuari té els privilegis necessaris. S'encarrega de mostrar tots els dossiers que compleixen els criteris introduïts i que no estan ni prestats ni tenen peticions en estat *PREPOK*, ja que en aquests dos casos no es poden traspasar els dossiers.

#### 6.6.1.41 La classe **TraspasarAhcDossier.java**

Igual que el préstec i recuperació de peticions es pot fer amb una pistola lectora de codis de barres, el traspàs de dossiers també.

Quan s'obre la finestra de traspàs de dossiers, se selecciona l'arxiu de destí i es llegeix l'etiqueta d'un dossier, s'executa aquesta acció.

Si l'arxiu de destí és *ART* o *AR2* (els dos arxius interns), l'únic que realitza aquesta acció és canviar l'arxiu del dossier, generant el corresponent registre a *AHC\_CAMBIOESTADO*.

En canvi, si el dossier de destí és *CLS*, el que es fa és canviar l'estat del dossier a *CLP*, que és un estat que indica que el dossier està pendent de ser enviat a l'arxiu extern, però no es canvia l'arxiu on es troba.

#### 6.6.1.42 La classe **EnviarAhcDossierACLS.java**

Igual que quan es demanen dossiers a l'arxiu extern s'ha d'enviar un fitxer indicant quins dossiers se sol·liciten, quan s'envien dossiers a l'arxiu CLS també s'ha d'enviar un altre fitxer indicant quins són els dossiers que viatjaran cap allà.

D'aquesta manera, l'usuari busca els dossiers que ha d'enviar al CLS (són els que tenen estat *CLP*), els selecciona i clica el botó d'enviar dossiers al CLS.

Si l'usuari té privilegis, es crea un fitxer que inclou els codis dels dossiers seleccionats i que es troben en estat *CLP*. Aquest fitxer, igual que els anteriors, s'envia a un directori determinat del servidor *cls.cspt.es*, on el personal de CLS l'anirà a buscar per a processar-lo i saber quins dossiers s'enviaran des de la CSPT.

A més, els dossiers que s'han inclòs al fitxer passen a estar en estat *CLT*, que indica que ja s'ha informat al CLS de que el dossier serà enviat.

#### 6.6.1.43 La classe **RecibirAhcDossierEnCLS.java**

Finalment, quan el personal de CLS processa el fitxer enviat i rep els dossiers enviats, genera un nou fitxer amb tots els dossiers que ha rebut, que també deixa al servidor *cls.cspt.es*.

Aquesta acció s'encarrega de processar el fitxer i posar tots els dossiers que conté en estat *CLS*, que indica que el dossier està a l'arxiu extern.

### 6.6.2 Les accions dels manteniments

Finalment, a continuació es descriuran quines classes Java es fan servir en les accions usades en els manteniments de les taules mestres.

Totes les taules mestres tenen els mateixos tipus de classe, que realitzen accions similars sobre les taules corresponents. És a dir, totes les taules tindran una classe que s'encarregarà de buscar les dades que hi ha a la taula (en realitat, les dades es troben al DM), una classe que s'executa quan es polsa el botó de crear un nou objecte del tipus seleccionat, una classe que executa l'acció de modificar o eliminar un objecte, etc.

Per aquest motiu, no s'explicaran totes les classes, sinó que es farà una explicació genèrica sobre cada tipus de classe, ja que el funcionament de totes les classes del mateix tipus és el mateix. Tal i com passa amb les classes descrites a l'apartat anterior, totes les classes d'aquest apartat deriven de la classe *Action.java* i implementen el mètode *perform()*.

Per altra banda, els noms de les classes són tots iguals, només canvien el final, que fa referència a la taula sobre la que treballen. Així, hi ha la classe *BuscarAhcArchivo.java* o la classe *BuscarAhcCategoria.java*, per exemple.

Cal dir que totes les accions, abans d'executar-se, comproven que l'usuari tingui privilegis per a fer-ho.

Les taules que tenen aquestes classes són les següents: **AHC\_ARCHIVO**, **AHC\_CATEGORIA**, **AHC\_DOSSCENTRO**, **AHC\_ERRORPREP**, **AHC\_ESTADODOSS**, **AHC\_MOT\_UBI**, **AHC\_MOTPRESTAMO**, **AHC\_SITUACION**, **AHC\_TIPODOSS** i **AHC\_UBICACION**.

#### 6.6.2.1 La classe **BuscarNomTaula.java**

Aquesta és l'acció que s'invoca quan es crida la pantalla de manteniment de la taula corresponent. Carrega per pantalla tot el contingut de la taula que toqui.

#### 6.6.2.2 La classe **NuevoNomTaula.java**

Acció que s'invoca quan, des del manteniment de la taula que s'estigui consultant, es polsa el botó de crear un nou registre. Aquesta acció s'encarrega de cridar a la pàgina JSP que conté el formulari per introduir les dades necessàries per a crear un nou objecte del tipus que sigui i el corresponent registre a la taula adient.



### **6.6.2.3 La classe *CrearNomTaula.java***

Un cop s'ha omplert el formulari amb les dades necessàries per a crear un nou objecte, es polsa el botó *Acceptar* de la pàgina, que crida aquesta acció, que és l'encarregada de crear el nou objecte i fer la inserció a la taula.

### **6.6.2.4 La classe *ModificarNomTaula.java***

Aquesta acció s'invoca quan, des del manteniment de la taula, i amb una fila de la pantalla seleccionada, es polsa el botó de modificar un nou registre. S'encarrega d'obtenir les dades de l'objecte que està marcat al panell i passar-les a la pàgina JSP que conté el formulari per a modificar l'objecte. D'aquesta manera, el formulari ja apareix carregat amb les dades que té l'objecte actualment i l'usuari només ha de modificar les que desitja.

### **6.6.2.5 La classe *ActualizarNomTaula.java***

Un cop s'han modificat les dades desitjades al formulari, es polsa el botó *Acceptar* de la pàgina, que crida aquesta acció, que és l'encarregada de modificar les dades que tenia l'objecte, fent a més la corresponent actualització a la taula.

### **6.6.2.6 La classe *EliminarNomTaula.java***

Amb una fila seleccionada, quan es polsa el botó d'eliminar, es crida aquesta acció que s'encarrega d'eliminar el registre que estigui marcat a la pantalla.

## **Capítol 7 Proves realitzades**

Un cop ja s'ha vist quina ha estat la implementació del mòdul, es descriuran quines han estat les proves que s'han realitzat al llarg del projecte.

Tot projecte ha de tenir una etapa de proves i tests, mentre s'està realitzant, per tal de verificar-ne el seu funcionament i si, això no és així, corregir els errors que s'hagin produït. De la mateixa manera, un cop acabat el projecte, s'ha de fer un test de proves finals abans de comercialitzar-lo o, en aquest cas, de posar-lo en funcionament a l'entorn de producció.

En primer lloc s'han realitzat diverses proves, durant la pròpia codificació del mòdul, per tal de garantir que cada acció està realitzant exactament allò per al que s'ha programat. Per posar un exemple, si es programa una acció que ha de crear un dossier, s'ha de comprovar que, en executar aquesta acció, realment s'estigui creant el dossier, informàticament parlant, amb les dades que li pertocuin. Aquestes proves han estat realitzades per part de l'equip professional d'HP per totes i cadascuna de les diferents accions que es poden realitzar al mòdul AHC.

Per altra banda, conforme s'ha anat acabant cada part del mòdul individualment (Gestió de Dossiers, per exemple), s'ha habilitat aquesta part en un entorn de proves. D'aquesta manera, els usuaris hi han tingut accés per tal de començar a familiaritzar-se amb la nova aplicació, i així també detectar possibles errors i/o proposar millores.

Mentre les primeres proves han consistit en fer tests individuals de cada acció, el personal d'arxiu s'ha encarregat de provar l'aplicació intentant simular quina seria la feina que realitzarien si l'aplicació ja estigués en funcionament. Així, simulant un circuit de treball real, s'han detectat errors que s'havien passat per alt en la primera fase de proves, ja que és pràcticament impossible contemplar totes les casuístiques a la primera. De la mateixa manera, també s'ha pogut veure quins punts de l'aplicació es podien millorar per tal d'agilitzar la interacció amb l'usuari.

Finalment, el personal de Sistemes de la CSPT s'ha encarregat de comprovar el rendiment del nou servidor, per tal d'assegurar que és capaç d'aguantar sobradament la càrrega de feina a la que es veu sotmès una vegada el mòdul AHC ja està en l'entorn de producció i hi estan tots els usuaris treballant. A la vegada, també han assegurat que els accessos a la base de dades són correctes i que no es provoquen bloquejos que puguin fer que l'aplicació es quedi penjada.

## **Capítol 8    Conclusions**

Una vegada ja s'ha fet tota la bateria de proves, es pot donar per finalitzada la codificació del mòdul AHC d'HCIS i ja se'n poden extreure les conclusions.

La idea de la realització del mòdul AHC era proporcionar a l'aplicació HCIS d'una part que li permetés gestionar Arxius d'Històries Clíniques d'hospitals, per aconseguir que fos una aplicació el més completa possible, tal i com s'ha pogut veure al primer capítol.

A més, també s'havia d'intentar que fos el més estàndard possible. S'ha de dir que aquest objectiu s'ha aconseguit en gran part, però que ha estat impossible fer-ho al 100%, ja que tot el que comporta el circuit d'enviament i recepció de dossiers amb l'arxiu extern (CLS) és quelcom molt específic de la CSPT, degut a que té el seu propi sistema de comunicació entre els arxius interns i l'extern. Així, aquesta part del mòdul només es farà servir a la CSPT, però la resta sí que és totalment estàndard i serveix per a treballar en qualsevol hospital.

Per altra banda, el mòdul AHC cobreix tots els requeriments que es van especificar inicialment, i que estan reflectits al tercer capítol, motiu pel qual es pot assegurar que aquest és un mòdul força complet, que permet realitzar i registrar tota la feina que s'ha de desenvolupar en un arxiu clínic.

Finalment, s'ha de comentar que el projecte ha sofert un endarreriment de dues setmanes respecte a la planificació inicial. Una part d'aquest temps de demora ha estat causat pels inconvenients que han anat sorgint durant el desenvolupament del mòdul, degut a errors que han anat sortint i a millores estètiques que han anat sol·licitant els usuaris després d'anar fent les seves proves. L'altre motiu d'endarreriment ha estat provocat per la manca d'aules d'informàtica on fer la formació als futurs usuaris. Com és obvi, sense tenir a tots els usuaris formats, s'ha hagut de retardar l'arrencament.

Amb el que s'ha vist en tots els punts anteriors, i considerant que el mòdul ja està en l'entorn de producció, es pot concloure que la realització del mòdul AHC s'ha finalitzat exitosament.

## **Capítol 9 Bibliografia**

### **9.1 Referències i articles**

- [1] <http://www.madrimasd.org/informacionidi/noticias/noticia.asp?id=38511&origen=notiweb>
- [2] <http://www.idc.com/spain/events/sanidad07.jsp>
- [3] <http://www.idg.es/computerworld/articulo.asp?id=159490>
- [4] [http://www.infochannel.es/articulo\\_item.php?numero=11419](http://www.infochannel.es/articulo_item.php?numero=11419)
- [5] [http://www.net2u.es/index.php/mod.noticias/mem.detalle/v\\_mem.listado/seccion.noticias/idnoticia.151/relcategoria.7/chk.56addc40e3bd0f019601544925d87800.html](http://www.net2u.es/index.php/mod.noticias/mem.detalle/v_mem.listado/seccion.noticias/idnoticia.151/relcategoria.7/chk.56addc40e3bd0f019601544925d87800.html)
- [6] <http://www.elcomerciodigital.com/gijon/20090328/aviles/agustin-dice-adios-papel-20090328.html>
- [7] <http://www.savac.es/gestor.php>
- [8] [http://www.acceso.com/display\\_release.html?id=1558](http://www.acceso.com/display_release.html?id=1558)
- [9] <http://java.sun.com/javase/6/webnotes/Itanium6u7.html>
- [10] <http://mindprod.com/jgloss/intellij.html>
- [11] <http://es.wikipedia.org/wiki/Oracle>
- [12] <http://es.wikipedia.org/wiki/Toad>
- [13] <http://www.syntevo.com/smartcvs/index.html>
- [14] <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport>
- [15] <http://www.asp.net/ajax/>

### **9.2 Altres web**

- [1] <http://java.sun.com>
- [2] <http://www.cica.es/formacion/JavaTut/Intro/tabla.html>
- [3] [http://personal.telefonica.terra.es/web/rlagob/java/java\\_inicio.htm](http://personal.telefonica.terra.es/web/rlagob/java/java_inicio.htm)

### **9.3 Llibres**

- [1] Jesús Bobadilla, Adela Sancho, *Comunicaciones y bases de datos con Java*. Ed.: Ra-Ma, 2003.
- [2] K. Arnold, J. Gosling, *El lenguaje de programación Java*. Ed.: Addison-Wesley/Domo, 1997.

## **Annex**

En aquest annex es mostrarà una imatge de les tres pantalles principals que formen el mòdul AHC (Gestió de Dossiers, Gestió de Peticions i Gestió de Préstecs) per tal d'explicar-ne breument el seu funcionament. Cal remarcar que les captures de pantalla són de l'aplicació en l'*Entorn de Desenvolupament* i que les dades del pacient que hi apareix són fictícies.

### **A.1 Pantalla de Gestió de Dossiers**

En la següent figura (Figura A.1.) es pot veure l'aspecte que té la pantalla de Gestió de Dossiers. Per accedir-hi, s'ha de triar l'opció *Arxiu* del menú esquerre de HCIS, i posteriorment la subopció *Gestió de Dossiers*.

Un cop oberta la pantalla, s'ha d'introduir el número d'història clínica del pacient per al qual es vol buscar la informació dels dossiers, i pulsar la tecla *Intro*. Si es desitja, també es pot fer servir la icona de la lupa que hi ha just al costat per a que s'obri la finestra de recerca de pacients, on es pot buscar el pacient per nom, cognoms, dni, etc.

Un cop seleccionat el pacient, a la part superior es mostren les dades principals del dossier que s'està consultant, com poden ser el tipus, la data de creació, la seva ubicació actual, etc.

A la part inferior hi ha dues parts diferenciades. A la part de l'esquerra hi apareixen tots els dossiers que té creats el pacient seleccionat, mentre que a la part dreta es mostra la informació sobre tots els préstecs i moviments que ha tingut aquest dossier, així com tots els episodis que ha tingut el pacient.

En funció del dossier que se seleccioni (si el pacient en té més d'un), la informació mostrada variarà, per tal de visualitzar les dades associades a la història clínica que s'està consultant. L'únic que no varia és la part de l'*Històric d'Episodis*, ja que això va lligat al pacient, que no ha canviat.

Finalment, a la part central de la pantalla hi ha la botonera, on hi ha els botons per a poder realitzar les accions de crear, modificar o eliminar dossier, imprimir etiquetes, prestar dossier, etc.

Hospital de Sabadell

Corporació

Parc Taulí

Localització de pacients

Historia Clínica

Resultats de Proves

Registres clínics

Hospitalització

Consultes Externes

Arxiu

Gestió de Dossiers

Gestió de Peticions

Pet. Espont. Urgents

Pet. Espont. NO Urgents

Gestió de Préstecs

Recuperació massiva

Traspàs massiu de Dossiers

Llistat d'històries en arxiu d'Ingressats

Llistat pendents d'Ingrés Taulí

Llistat HHCC d'AR2 amb CEJ en els propers dies

Llistat PINS Sta Fe

Llistat HHCC recuperades

Llista Negra Prearxiu

Llistat per terminació

Llistat CEX

Llistat Exteriors

Gestió de Dossiers

Crític de recerca

Pacient

NHC 82943

Prova Prova, Prova

Tipus: 0 Historia General

Ubicació: AR2 - Arxiu Edifici Santa Fe

Estat: Disponible

Situació: Actiu

D. de creació: 02/07/2008

Último mov.: 10/03/2009

Electrónica: ☐

Volums: 1

Observacions:

Dossiers

Codi L 82943

Historia de préstamos

Historia de movimientos

Historia de episodios

Data T	Motivo	Centre	Servei	Data Dev.	Q. Infermeria	Agenda	Netge	Ubicació	Obs.
05/02/2009 10:44:52	Codificació Santa Fe	Hospital de Sabadell		05/02/2009					
21/01/2009 14:44:19	Auditoria	Hospital de Sabadell		21/01/2009					
07/01/2009 11:08:21	Consultes Externes	Hospital de Sabadell Oncologia		21/01/2009	SAIGI			CTD - Consulta Taulí Sala D	
05/01/2009 10:41:29	Auditoria	Hospital de Sabadell		05/02/2009					
03/12/2008 11:23:59	Estudi	Hospital de Sabadell		18/12/2008					estud
13/10/2008 10:45:37	Particular	Hospital de Sabadell		05/02/2009				Gallach Montero, Marta	prueb
15/09/2008 13:28:44	Estudi	Hospital de Sabadell		15/09/2008				Gil Prades, Montserrat	ES UN
01/09/2008 13:53:37	Ingrés	Hospital de Sabadell		09/10/2008				ALB - Albada	

1 dossiers

27 préstecs

Figura A.1. Pantalla de Gestió de Dossiers

## A.2 Pantalla de Gestió de Peticions

La següent figura (Figura A.2.) mostra com és la pantalla de Gestió de Peticions. Per obrir-la, s'ha de triar l'opció *Arxiu* del menú esquerre de HCIS, i posteriorment la subopció *Gestió de Peticions*.

La pantalla de Gestió de Peticions està dividida en dues parts. La meitat superior de la pantalla està formada per un formulari per a introduir les dades per la que es vol filtrar la cerca de peticions, com poden ser el pacient, el motiu de sol·licitud, l'estat en què es troba la petició, la data de petició, etc.

És molt important introduir filtres a l'hora de realitzar cerques de peticions, ja que el volum d'aquestes es tan gran que, sinó s'acota la cerca, aquesta seria molt lenta. De fet, no es permet realitzar cerques de peticions sense introduir filtres, ja que suposaria consumir molts recursos del servidor, fent que el rendiment de l'aplicació es veiés afectat.

A més, aquesta pantalla també es fa servir per a preparar les peticions (comprovar on es troba cada dossier per a saber si es pot prestar o no). Quan es clica el botó *Preparar*, es preparen totes les peticions que compleixen els criteris establerts al formulari. Quan es fa la preparació no es realitza una cerca posterior de peticions, ja que el volum de peticions que es prepara acostuma a ser força gran, i fer una cerca i mostrar els resultats per pantalla comportaria els problemes que s'acaben d'explicar.

La meitat inferior està formada per una botonera i el panell on es mostren els resultats de la recerca.

Les accions que es poden fer des de la botonera s'executen sobre la fila (petició) que està seleccionada. Hi ha accions que es poden fer només sobre una fila, com pot ser la de modificar petició, accions que es poden fer sobre diverses files, eliminar peticions, per exemple, i accions que no han de tenir cap fila seleccionada, com són la de crear una nova petició o obrir la finestra per a prestar peticions amb la pistola lectora de codis de barres.

87



### A.3 Pantalla de Gestió de Préstecs

Finalment, a la Figura A.3. hi ha una visualització de com és la pantalla de Gestió de Préstecs. Per obrir-la, s'ha de triar l'opció *Arxiu* del menú esquerre de HCIS, i posteriorment la subopció *Gestió de Préstecs*.

Com es pot observar, la pantalla de Gestió de Préstecs és molt similar a la de Gestió de Peticions. També està dividida en dues parts, tenint, a la meitat superior de la pantalla, un formulari per a introduir les dades per filtrar la cerca de préstecs, com poden ser el pacient, el motiu de préstec, la data de préstec, etc. S'ha de considerar que només es mostren els préstecs actius. Tot i que no és tant important filtrar les cerques de préstecs, degut a que el nombre de préstecs actius és molt inferior al de peticions, sí que és aconsellable. De fet, tampoc està permès buscar préstecs sense introduir cap filtre.

Igual que a la pantalla de Gestió de Peticions, la meitat inferior de la pantalla de Gestió de Préstecs també està formada per una botonera i el panell on es mostren els resultats de la recerca.

Les accions que es poden fer des de la botonera s'executen sobre la fila (préstec) que està seleccionada. Hi ha accions que es poden fer només sobre una fila, com pot ser la de modificar préstec, accions que es poden fer sobre diverses files, recuperar préstecs, per exemple, i accions que no han de tenir cap fila seleccionada, com la de crear un nou préstec.

[illegible]

Figura A.3. Pantalla de Gestió de Préstecs

**Resum:**

El sector sanitari és un dels més importants i sensibles dintre de l'economia espanyola, ja que és un element clau del benestar social. Per aquest motiu, el sector de la sanitat s'enfronta a nombrosos reptes, i les Tecnologies de la Informació i les Comunicacions (TIC) obren un ampli ventall de possibilitats de renovació i millora.

És per això que Hewlett-Packard està desenvolupant una aplicació informàtica per a hospitals, anomenada HCIS (Health Care Information System).

El projecte que es presenta en aquesta memòria és el desenvolupament del mòdul d'HCIS per a gestionar l'Arxiu d'Històries Clíiques d'un hospital, i la seva implantació a la Corporació Sanitària Parc Taulí de Sabadell.

**Resumen:**

El sector sanitario es uno de los más importantes y sensibles dentro de la economía española, ya que es un elemento clave del bienestar social. Por este motivo, el sector de la sanidad se enfrenta a numerosos retos, y las Tecnologías de la Información y las Comunicaciones (TIC) abren un amplio abanico de posibilidades de renovación y mejora.

Es por eso que Hewlett-Packard está desarrollando una aplicación informática para hospitales, llamada HCIS (Health Care Information System).

El proyecto que se presenta en esta memoria es el desarrollo del módulo de HCIS para gestionar el Archivo de Historias Clínicas de un hospital, y su implantación en la Corporació Sanitària Parc Taulí de Sabadell.

**Abstract:**

The health sector is one of the most important and sensitive issues within the Spanish economy as a key element of social welfare. For this reason, the health sector faces many challenges, and the Information and Communications Technologies (ICT) open up a wide range of possibilities for renewal and improvement.

Because of this, Hewlett-Packard is developing a computer application for hospitals, named HCIS (Health Care Information System).

The project presented in this memory is the HCIS development module for managing the Medical Records of a hospital, and its implantation at Corporació Sanitària Parc Taulí from Sabadell.